



KERNFORSCHUNGSANLAGE JÜLICH GmbH

Zentralinstitut für Angewandte Mathematik

**Simulationsmodell für den Dateitransfer
zwischen Großrechnern auf der Basis
des Hyperchannel/50**

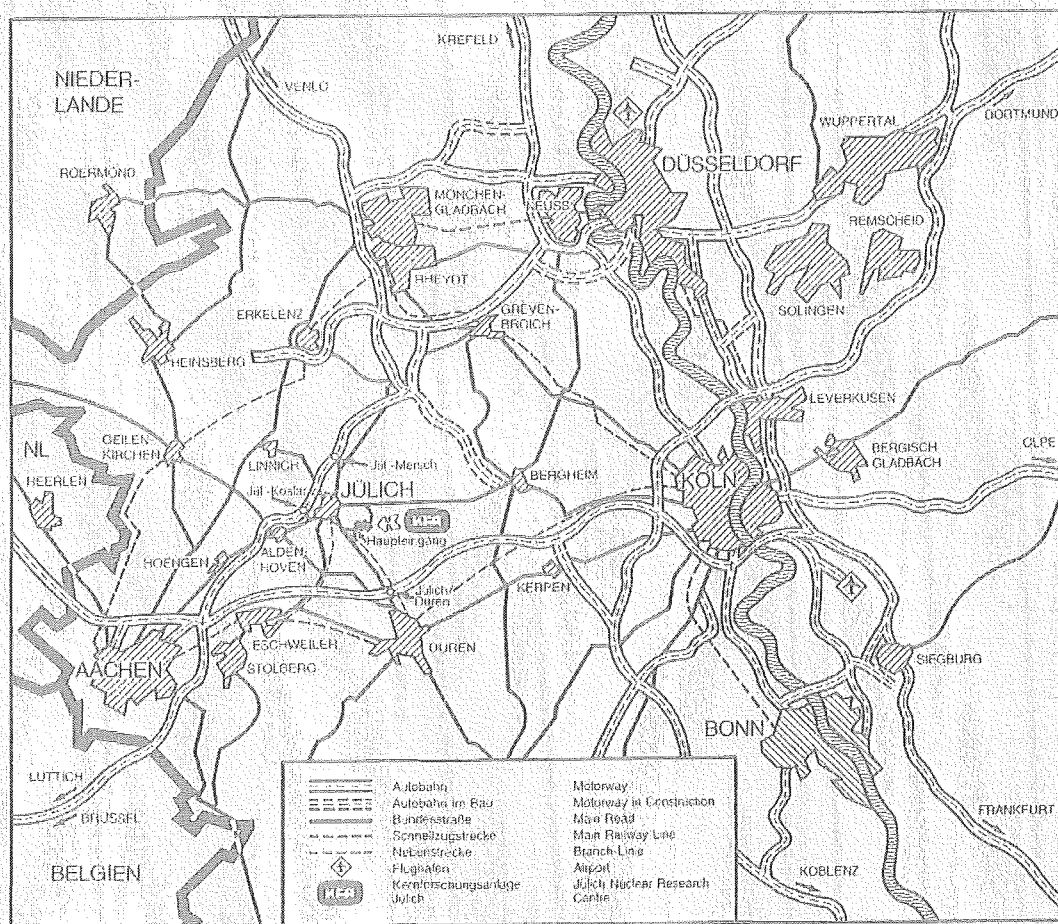
von

Richard Geibel

Jüli - Spez - 356

Juni 1986

ISSN 0343-7639



Als Manuskript gedruckt

Spezielle Berichte der Kernforschungsanlage Jülich – Nr. 356

Zentralinstitut für Angewandte Mathematik Jül - Spez - 356

Zu beziehen durch: ZENTRALBIBLIOTHEK der Kernforschungsanlage Jülich GmbH

Postfach 1913 · D-5170 Jülich (Bundesrepublik Deutschland)

Telefon: 02461/610 · Telex: 833556-0 kf d

Simulationsmodell für den Dateitransfer zwischen Großrechnern auf der Basis des Hyperchannel/50

von

Richard Geibel

VORWORT

Die vorliegende Diplomarbeit wurde am Institut für Informatik IV der RWTH Aachen bei Herrn Prof. Dr. O. Spaniol im Zentralinstitut für Angewandte Mathematik der KFA Jülich angefertigt. Für das Interesse an meiner Arbeit sowie vieler stimulierender Hinweise im Verlauf dieser Arbeit möchte ich mich bei Herrn Prof. Dr. O. Spaniol bedanken.

Mein Dank gilt weiterhin dem Direktor des Zentralinstituts für Angewandte Mathematik, Herrn Dr. F. Hoßfeld, der meine Arbeit in diesem Institut ermöglichte und mir großzügige Hilfen durch die Unterstützung vieler Institutsmitarbeiter zukommen ließ.

An dieser Stelle muß in besonderer Weise die Unterstützung durch Herrn W. Gürich erwähnt werden, der mit großem Interesse meine Arbeit im Zentralinstitut für Angewandte Mathematik verfolgte und durch dessen Anregungen und Hinweise diese Diplomarbeit entstehen konnte.

INHALTSVERZEICHNIS

1.0	Einleitung	1
2.0	Problembeschreibung	3
2.1	Aktuelle Situation	3
2.2	Angestrebte Situation	5
3.0	Kommunikation	9
3.1.1	Hardware	10
3.1.2	Protokoll	13
3.1.2.1	Frames und Framesequenzen	18
3.1.2.2	Wait-Flip-Flop	25
3.1.2.3	Virtuelle Adapter-Verbindungen	25
3.1.2.4	Multi-Trunk Networks	30
3.1.3	Software	32
4.0	Modell	35
5.0	Simulation	39
6.0	Programm	45
6.1	Programmstruktur	46
6.2	Eingabe	48
6.3	Ausgabe	50
7.0	Fallstudien	61
7.1	Gleiche Kanalgeschwindigkeit	61
7.2	Unterschiedliche Kanalgeschwindigkeit	62
7.3	Half-Duplex Übertragung	65
7.4	Warteschlangenbenutzung der höheren Ebene	67
7.5	Overhead bei der Übertagung mehrerer Dateien	69
7.6	Deadlock Auflösung	73
7.7	Nutzen des Trunks bei zwei parallelen Übertragungen	77
8.0	Ergebnisse	79

8.1	Auswirkung unterschiedlicher Kanalgeschwindigkeit	79
8.2	Effektive Datenrate	80
8.3	Trunk-Auslastung	84
8.4	End-Delay	85
8.5	Antwortzeit	89
9.0	Diskussion	97
10.0	Literatur	101

VERZEICHNIS DER ABBILDUNGEN

Abbildung 1.	Aktuelle Situation	4
Abbildung 2.	Angestrebte Situation: Schritt 1	6
Abbildung 3.	Angestrebte Situation: Schritt 2	7
Abbildung 4.	Prozessor-Adapter Aufbau	11
Abbildung 5.	Parallele Arbeitsweise der Adapterpuffer	13
Abbildung 6.	Zeitphasen des CSMA/CA-Verfahrens	15
Abbildung 7.	Trunk-Zugriffs-Verfahren	18
Abbildung 8.	Ablauf einer Message-Only-Sequenz	20
Abbildung 9.	Transmisson Frames	22
Abbildung 10.	Zeiten für eine Message-Only-Framesequenz	24
Abbildung 11.	Adapter-Reserve-Loop	27
Abbildung 12.	Multi-Trunk-Konfiguration	30
Abbildung 13.	ISO-OSI Referenzmodell	32
Abbildung 14.	Network-Exchange und Bulk-File-Transfer	34
Abbildung 15.	Warteschlangenmodell für das Netz	36
Abbildung 16.	Message-With-Data-Sequenz	37
Abbildung 17.	Ankunftsrate der Ereignisse	41
Abbildung 18.	Ankunftsrate der Ereignisse	42
Abbildung 19.	Warteschlangenmodell und OSI-Layer	45
Abbildung 20.	Eingabe -> Programm -> Ausgabe	47
Abbildung 21.	Eingabedaten für das Simulationsprogramm	49
Abbildung 22.	Statistik eines Simulationslaufes	51
Abbildung 23.	Antwortzeit, Bedienzeit und Wartezeit	53
Abbildung 24.	Prozentuale Aufspaltung in Bedien- und Wartezeit	54
Abbildung 25.	Ereignisdiagramm für den Host	55
Abbildung 26.	Zustände des Programms	58
Abbildung 27.	Ereignisdiagramm für den Adapter	60
Abbildung 28.	Sequenz bei gleicher Kanalgeschwindigkeit	62
Abbildung 29.	Unterschiedliche Kanalgeschwindigkeit: 1. Fall	63
Abbildung 30.	Unterschiedliche Kanalgeschwindigkeit: 2. Fall	64
Abbildung 31.	Half-Duplex Übertragung	66
Abbildung 32.	Warteschlangenbenutzung der höheren Ebene	68
Abbildung 33.	Übertragung mehrerer Dateien: Idle	71
Abbildung 34.	Übertragung mehrerer Dateien: Retry	72

Abbildung 35. Two-Adapter-Reservation-Request-Loop	74
Abbildung 36. Three-Adapter-Reservation-Request-Loop	76
Abbildung 37. Zwei parallele Übertragungen	78
Abbildung 38. Kanalgeschwindigkeit <-> Kanalzeit	79
Abbildung 39. Kurven für die Kanalverzögerung	80
Abbildung 40. Maximale Effektive Datenrate [Mbit/s]	81
Abbildung 41. Effektive Datenrate [Mbit/s] <-> Kanalgeschwindigkeit	83
Abbildung 42. Trunk-Auslastung <-> Effektive Datenrate	84
Abbildung 43. Ausbreitung des Signals im Trunk	86
Abbildung 44. Modifiziertes End-Delay	88
Abbildung 45. Antwortzeit bei einem lasterzeugenden Paar	90
Abbildung 46. Verzahnung von zwei Sequenzen	91
Abbildung 47. Aufbau der Versuche zur Messung der Antwortzeit . . .	92
Abbildung 48. Antwortzeit bei zwei lasterzeugenden Paaren	93
Abbildung 49. Antwortzeit <-> Trunk-Auslastung	95

1.0 EINLEITUNG

Eine stark wachsende Spezialisierung der zu lösenden mathematischen und technischen Probleme sorgte für eine zunehmende Spezialisierung der benutzten Rechnersysteme. Um weiterhin eine Kommunikation der unterschiedlichen Rechnersysteme untereinander zu ermöglichen, besteht die Notwendigkeit, die einzelnen Maschinen zu verbinden.

Der Zusammenschluß verschiedenartiger Rechnersysteme sowie der notwendigen Peripheriegeräte kann mit Hilfe eines Netzwerkes realisiert werden. Das Netz ermöglicht den Daten- und Informationsaustausch der Rechner untereinander, sowie die gemeinsame Nutzung von Ressourcen.

Besonders Programme und Dateien wie Datenbanken, Compiler, Libraries, etc. sollten zur Vermeidung von Redundanz und Inkonsistenz zentral verwaltet werden. Dazu bietet sich ein in das Netz integrierter File Server an.

Die Untersuchung der Kommunikationsweise mehrerer Großrechner zum Austausch von Dateien auf der Basis eines Netzes soll Gegenstand dieser Arbeit sein.

Betrachtet wird die Konfiguration eines großen technisch-wissenschaftlichen Rechenzentrums, welches über mehrere Universalrechner, einen schnellen Vektorrechner und über ein Verbindungsnetzwerk verfügt. Die Grundlage dieses Aufbaus bildet das Rechnersystem im Zentralinstitut für angewandte Mathematik der Kernforschungsanlage in Jülich.

Untersucht wird der Zusammenschluß der einzelnen Anlagen zu einem Gesamtsystem. Insbesondere sollen dabei der Durchsatz und die Antwortzeiten des Netzes und seiner Komponenten näher betrachtet werden.

Durch die Abbildung der betrachteten Konfiguration in ein Simulationsmodell sollen Aussagen über das Netzverhalten gewonnen werden und Erkenntnisse für eine günstige Wahl der Parameter erzielt werden.

2.0 PROBLEMBESCHREIBUNG

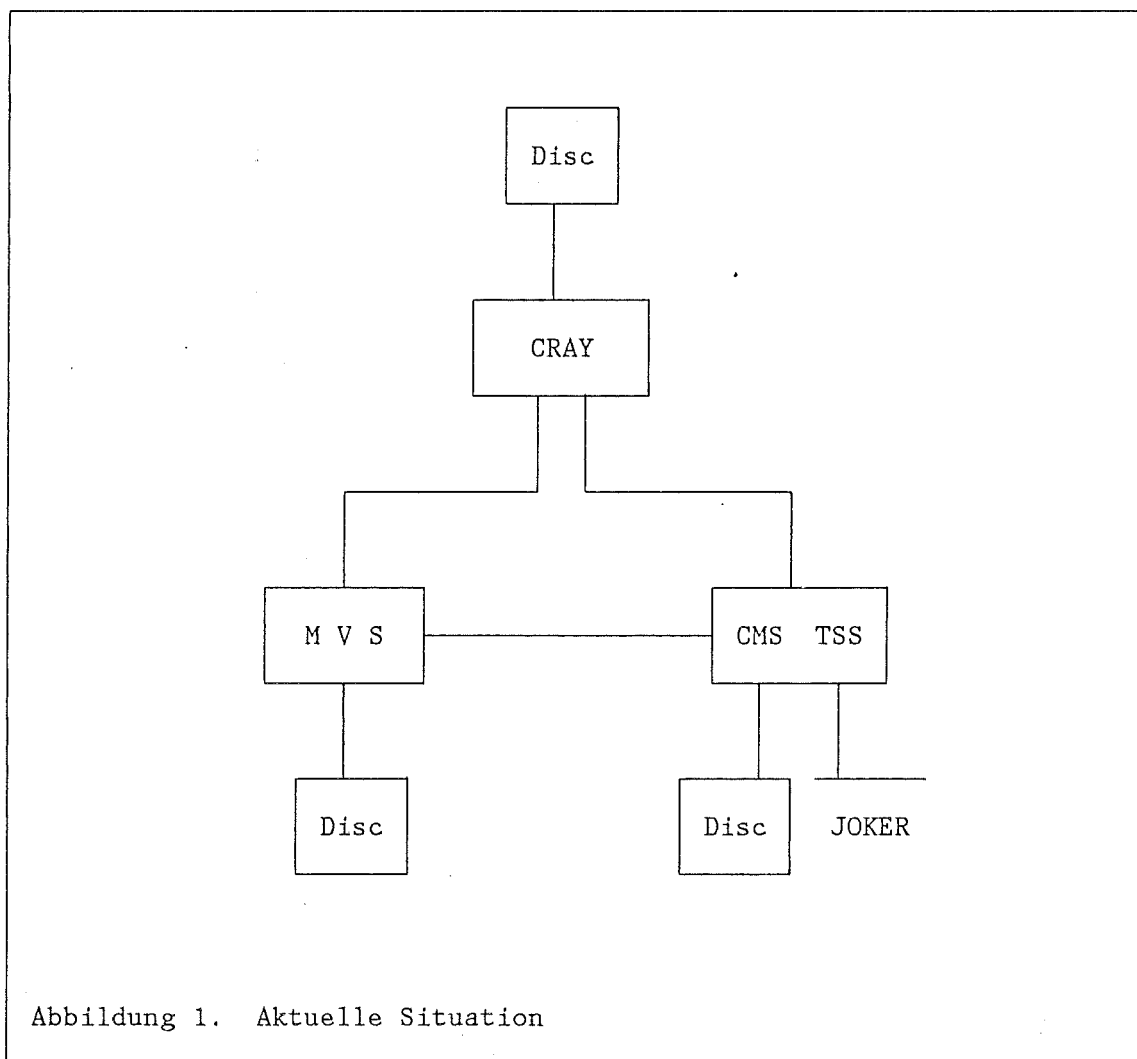
Im folgenden Kapitel soll zunächst durch Beschreibung der aktuellen Rechenzentrumssituation eine Bestandsaufnahme durchgeführt werden. Dabei wird sowohl die Rechnerseite als auch die Kommunikationsseite analysiert.

Nach dieser Schilderung des Status Quo soll die anzustrebende zukünftige Situation, unter Berücksichtigung der notwendigen Komponenten, erläutert werden. Hierbei wird insbesondere auf die zentrale Rolle des Kommunikationsnetzes eingegangen.

2.1 AKTUELLE SITUATION

Das betrachtete Rechenzentrum der KFA verfügt zur Zeit über folgende Großrechneranlagen (Abbildung 1 auf Seite 4):

1. CRAY X-MP 22 mit dem Betriebssystem 'Cray Operating System' (COS). Als schneller Vektorrechner wird die Cray vorrangig zur Lösung rechenintensiver technisch-wissenschaftlicher Aufgaben benutzt.
2. IBM 3081 D32 unter dem Betriebssystem 'Virtual Machine' (VM) in Verbindung mit CMS zur Bearbeitung interaktiver Aufgaben. Zusätzlich wird unter VM das Timesharing-Betriebssystem TSS betrieben, welches als Endstelle für das Experimentrechnerkopplungssystem JOKER benutzt wird.
3. IBM 3081 D32 für Stapelverarbeitung unter dem Betriebssystem 'Multiple Virtual Storage' (MVS).



Im Mittelpunkt der verschiedenen Rechnersysteme steht der Verbund der CRAY mit den beiden IBM 3081 /KFA 84/. Diese enge Kopplung ist notwendig, da eine direkte Benutzung der Cray nicht möglich ist. Aufgrund des speziellen Einsatzbereiches der Cray als Hochleistungsrechner ist kein unmittelbarer Zugriff der Benutzer über ihre Terminals vorgesehen. Statt dessen werden sogenannte Front-End-Rechner zur Bedienung der Cray benötigt. Sie übernehmen den gesamten I/O Verkehr, d.h. die Bereitstellung der benötigten Daten, die Aufbereitung der Jobsteuersprache und den Empfang der Ergebnisse. Diese Aufgabe fällt den beiden IBM 3081 als Front-End-Rechner für den Vektorcomputer zu. Die seitens der Cray unterstützte Anschlußmöglichkeit von Rechnern mit unterschiedlichen Betriebssystemen erlaubt die Benutzung der Cray sowohl von der Batch-Seite (MVS), als auch von der Dialog-Seite (VM/CMS) aus. Die Kommunikation erfolgt über eine direkte Kanal- zu Kanalverbindung, mittels eines von der Firma CRAY erstellten

Programmes. Diese 'Cray-Station-Software'-Programmpakete laufen auf den IBM-Anlagen als Erweiterung der jeweiligen Betriebssysteme und bilden die Grundlage für den Rechnerverbund.

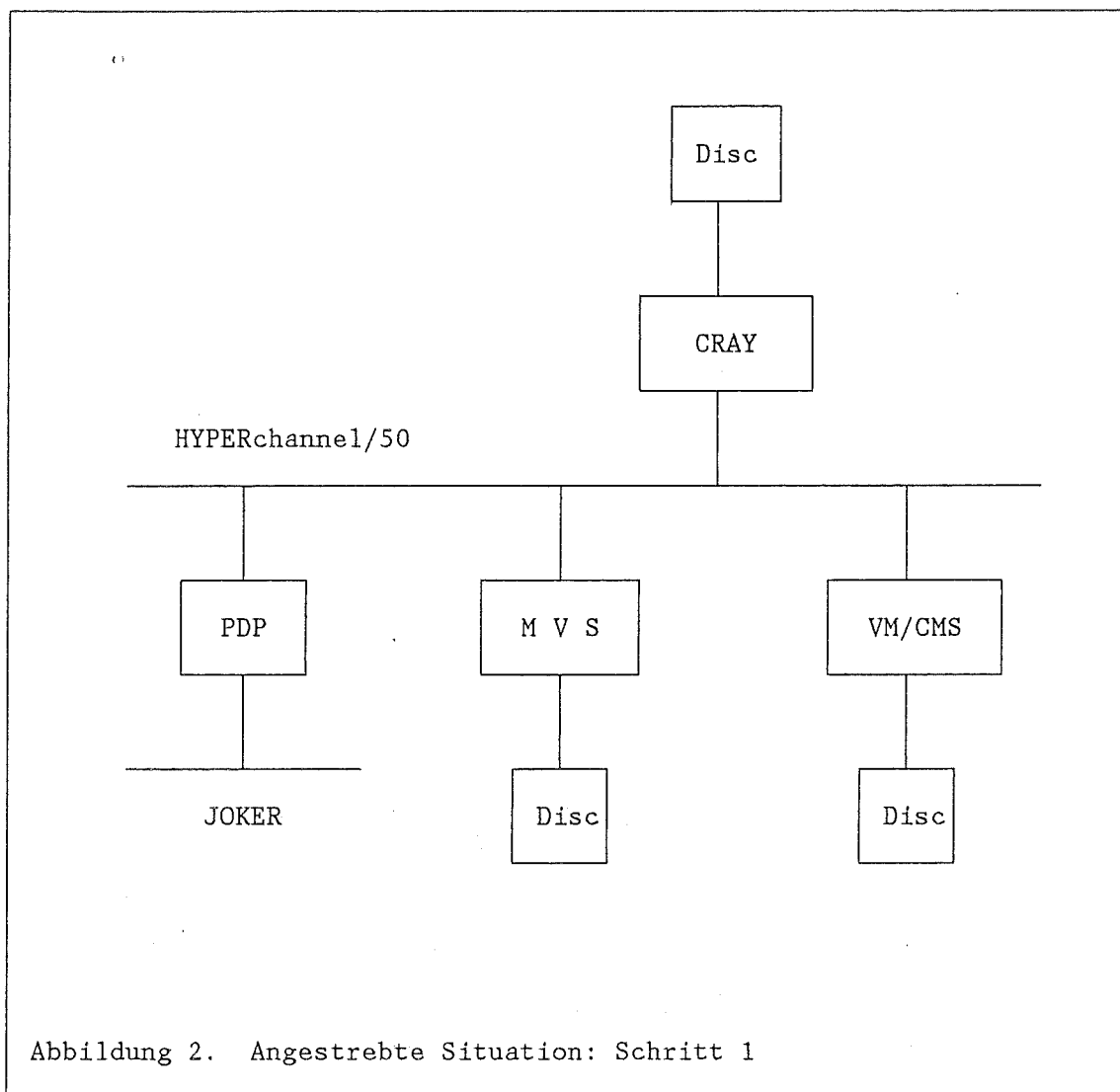
Zusätzlich sind die beiden IBM 3081 miteinander verbunden. Auch hierbei erfolgt der Datenaustausch direkt über eine schnelle Kanal- zu Kanalverbindung. Auf der Batch-Maschine ist ein 'Job Entry Subsystem' (JES 2) und auf dem Dialogrechner ein 'Remote Spooling Communications Subsystem' (RSCS) implementiert, welche die Kommunikation zwischen den beiden Computern abwickeln. Auf diese Weise ist die Möglichkeit des File-Transfers und der Job-Submission von einem Betriebssystem zu dem anderen geschaffen worden.

Das Timesharing-System TSS dient der Kopplung von Experimentrechnern unterschiedlichen Fabrikate in den einzelnen Instituten der KFA mit den zentralen Großrechnern. Der Anschluß an die IBM 3081 erfolgt mit Hilfe des bereits 1972 installierten JOKER-Netzes. Verbunden werden mit diesem Netz zur Zeit 48 Computer, wobei 15 mit einer 'hohen' Geschwindigkeit von 1.44 Mbit/s und 33 mit einer langsameren Geschwindigkeit von 9600 Baud übertragen können. Das Joker-System bietet den Benutzern mittels 'Remote Job Entry' (RJE) die Möglichkeit des File-Transfers von und zu ihren Experimentrechnern.

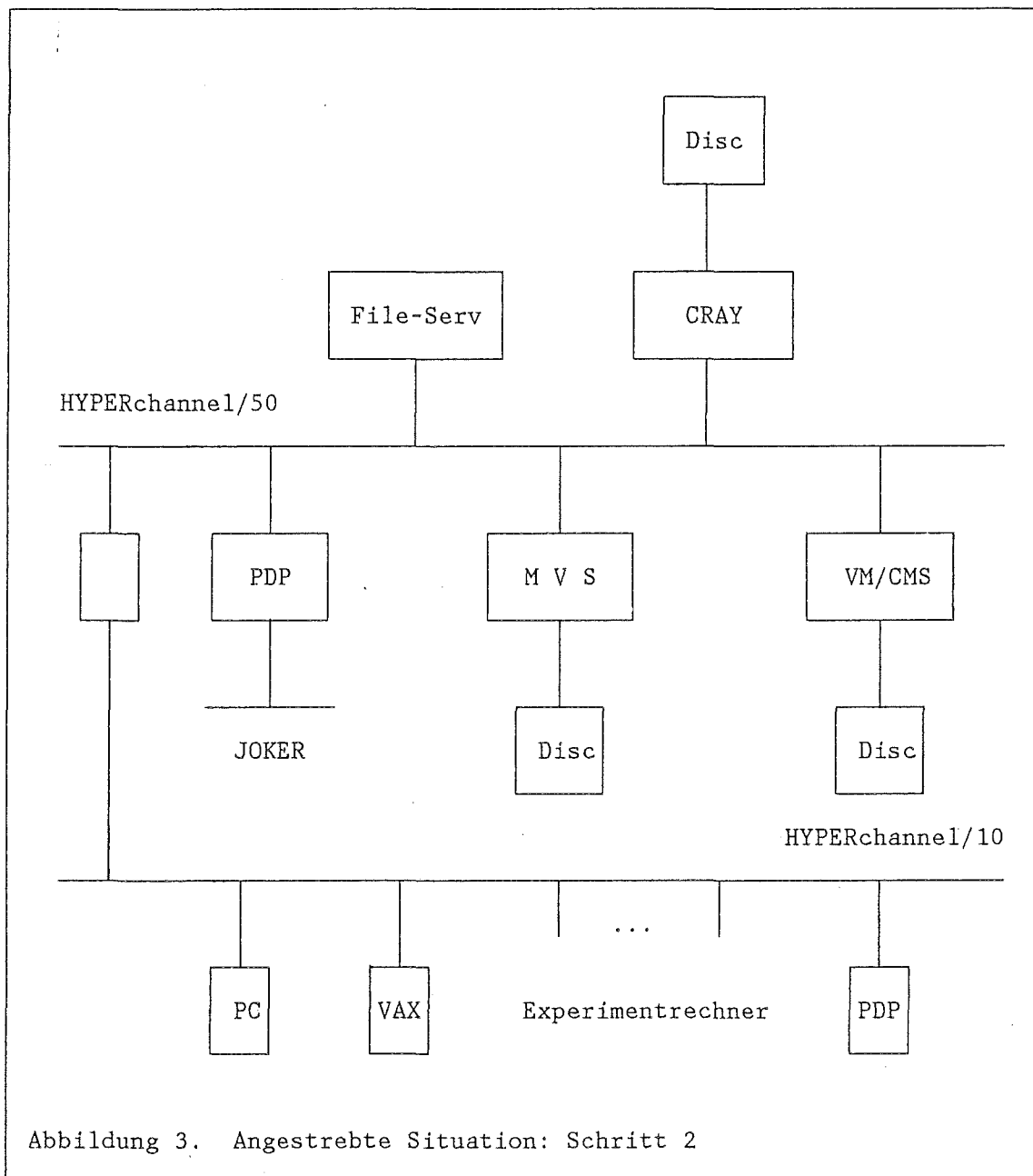
Diesen bestehenden Verbund kann man aufgrund seiner Architektur als 'vermaschtes Netz' bezeichnen, wobei nur in sehr eingeschränktem Maß Netzfunktionen unterstützt werden. Im wesentlichen existiert die Möglichkeit Jobs zu submittieren, also eine reine Batch Anwendung (Abbildung 1 auf Seite 4).

2.2 ANGESTREBTE SITUATION

Das Ziel der geplanten und teilweise bereits realisierten Arbeiten ist die bestehenden Rechner an ein gemeinsames Netz anzuschließen und einen File-Server zu integrieren.



Der erste Schritt zu dieser Konfiguration ist die Verbindung der CRAY und der beiden IBM 3081 mit Hilfe eines geeigneten 'High-Speed-Networks'. Die bisherige Kanal- zu Kanalverbindung und die dafür benötigte Kommunikationssoftware (Cray Station Software, JES 2, RSCS) innerhalb dieser drei Mainframes wird somit durch das Netz abgelöst. Als Netzwerk wurde der HYPERchannel/50 gewählt (siehe dazu das Kapitel "Kommunikation" auf Seite 9). Das Betriebssystem TSS, welches auf dem IBM-Rechner 3081 (VM/CMS) abläuft, wird nicht mehr in diesen Verbund integriert, sondern seine bisherige Funktion als Kommunikationsschnittstelle zwischen zwei unterschiedlichen Netzen wird durch einen kleinen Verbindungsrechner (PDP 11/84) übernommen (Abbildung 2).



Als zweite Stufe der Arbeiten ist die Entlastung des Joker-Netzes durch ein 'Inhouse'-Netz mit einer höheren Übertragungsrate und einer moderneren Technologie geplant. Hierfür ist der HYPERchannel/10 vorgesehen worden (siehe Kapitel "Kommunikation" auf Seite 9). Während einer Übergangsphase werden beide Netze parallel betrieben. Der HYPERchannel/10 soll die Institute der KFA mit dem zentralen Rechenzentrum verbinden und eine Kommunikation der vorhandenen Experimentrechner mit den Mainframes erlauben (Abbildung 3).

Eine solche Netzstruktur würde den Einsatz eines File-Servers, der sowohl von den Großrechnern, als auch von den über das 'Inhouse'-Netz angeschlossenen Rechnern genutzt werden könnte, ermöglichen. Neben dem File-Server, mit dem die Vorteile der Datenkonsistenz und der Vermeidung von Datenredundanz verbunden sind, bietet sich die Nutzung weiterer gemeinsamer Ressourcen wie der Printer, des Plotters und des Mikrofilm-Plotters an.

3.0 KOMMUNIKATION

Die Anforderungen an das Kommunikationsnetz werden durch die besondere Struktur und den speziellen Einsatzbereich des Rechenzentrums der KFA bestimmt. Insbesondere die Verwendung mehrerer Großrechner (CRAY, IBM) und der gewünschte Einsatz eines File-Servers bestimmen den Charakter des einzusetzenden Netzes.

Die wesentliche Aufgabe des Netzes wird die Ermöglichung der Kommunikation aller angeschlossenen Rechner und Peripheriegeräte sein. Darüber hinaus soll es über Gateway-Möglichkeiten zu anderen Netzen, insbesondere zu einem Netz in dem Geschwindigkeitsbereich von 10 Mbit/s, verfügen.

Da eine sehr schnelle Übertragung der Daten von Computer zu Computer bzw. File-Server erforderlich ist, um die bestehenden Responsezeiten zu verbessern oder zumindest beizubehalten, muß das Netz mit einer hohen Übertragungsgeschwindigkeit arbeiten. Bestimmt wird diese Geschwindigkeit durch den Anschluß der Netzadapter direkt an die Kanäle der einzelnen Maschinen. Zur Zeit liegt die Kanalgeschwindigkeit bei den IBM-Rechner bei 1.5 bzw. 3 Mbyte/s und bei der CRAY X-MP bei 6 Mbyte/s. Daraus ergibt sich für das Netz eine minimal wünschenswerte Übertragungsgeschwindigkeit von 12 Mbit/s; jedoch könnte eine Rate von 24 Mbit/s und sogar von 48 Mbit/s durch die angeschlossenen Rechner erreicht werden.

Die Entscheidung im Rechenzentrum der KFA als 'Very-High-Speed-Network' den HYPERchannel/50 der Firma Network Systems Corporation einzusetzen fiel aus mehreren Gründen:

- es handelt sich bei diesem Netz um ein offenes System im Sinne seiner Nutzungsmöglichkeit durch Rechner verschiedener Hersteller;
- es existiert die Software der Ebenen 1 bis 5 (Netex) für die unterstützten Rechnertypen (siehe "Software" auf Seite 32);
- im Bereich der Großrechnervernetzung existiert zur Zeit kaum eine Alternative zum HYPERchannel/50.

- von der selben Firma wird der HYPERchannel/10 angeboten, der im Geschwindigkeitsbereich von Ethernet liegt und über die gleiche Software (Netex) verfügt. Zusätzlich besteht die Möglichkeit an den HYPERchannel/10 direkt IBM Bildschirme anzuschließen und über bestehende Steckkarten IBM PC's.

In der weiteren Arbeit wird der HYPERchannel/50 betrachtet.

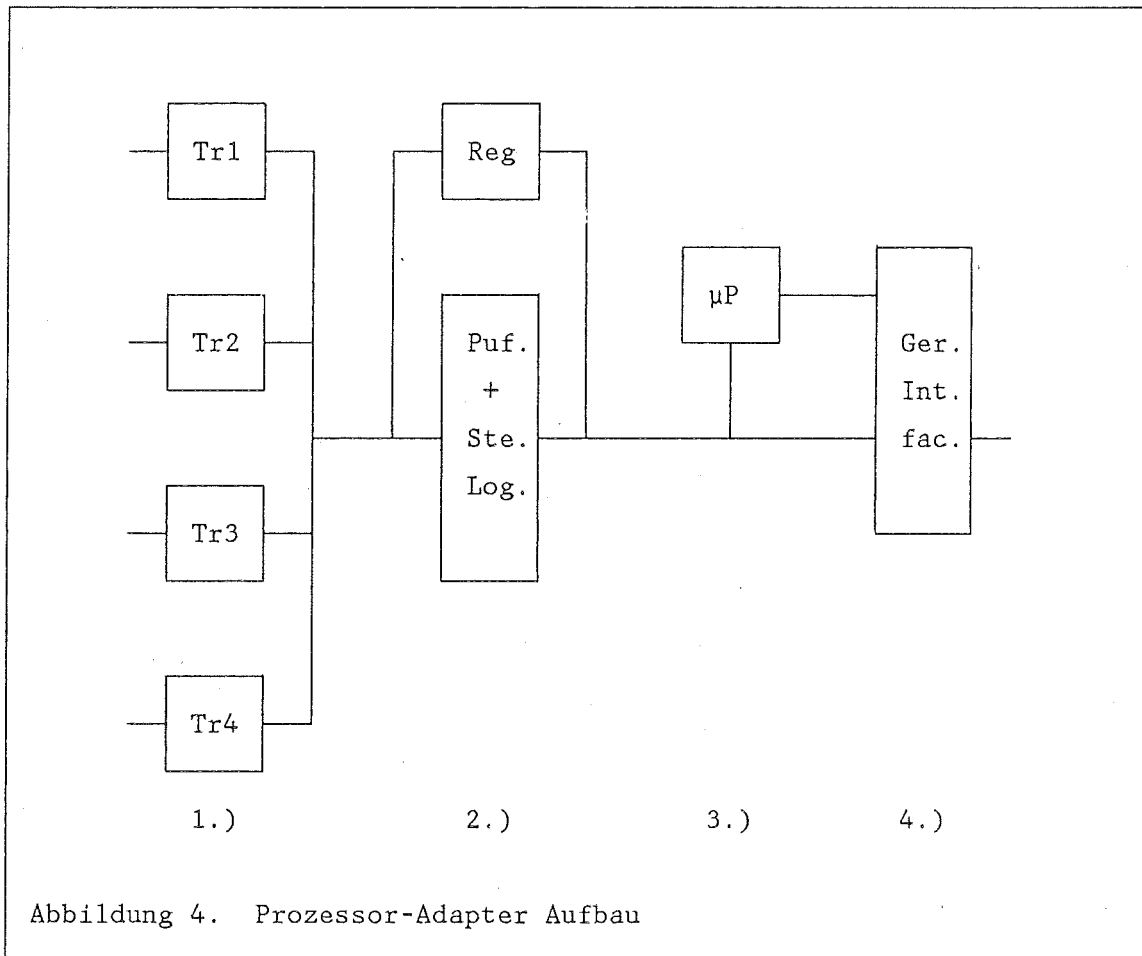
3.1.1 Hardware

Der HYPERchannel/50 von der Firma Network Systems Corporation wurde speziell für den Bereich der Mainframekopplung entworfen /NSC 85/. Er arbeitet mit einer Geschwindigkeit von 50 Mbit/s mittels Basisbandübertragung und wurde konzipiert, um eine in der Regel geringe Anzahl von Rechnern und Peripherieeinheiten zu verbinden. Bei der Architektur handelt es sich um ein Bussystem, welches als gemeinsam benutztes Medium Koaxialkabel verwendet.

In HYPERchannel/50 Netzwerken können verschiedenartige Maschinen, welche mit einer hohen Übertragungsgeschwindigkeit arbeiten (Mainframes, Server, etc.) mittels einer oder mehrerer Leitungen verbunden werden. Mit Hilfe von Interfaceeinheiten, die auch Adapter genannt werden, erfolgt der Anschluß der einzelnen Anlagen an die ein bis vier passiven Koaxialkabel. Solche Adapter sind inzwischen für die meisten Rechnertypen entwickelt worden, und sind insbesondere für die in der KFA eingesetzten Computer (CRAY, IBM, DEC) verfügbar. Die Adapter werden durch Verbindungsstecker an welche die Leitungen angeschraubt werden, mit dem Trunk verbunden (und nicht geklemmt wie bei Ethernet). Der minimale Abstand zwischen zwei Adaptern wird durch die kleinste erlaubte Differenz zwischen den Prioritäten-Delays bestimmt, und beträgt ca. 13 Meter (40 ft) /Fran 82/. Das Koaxkabel (auch Trunk genannt) läßt bei der verwendeten Übertragungsgeschwindigkeit eine Ausbreitungsgeschwindigkeit des Signals von 4 ns/m bzw. 1.32 ns/ft zu.

Jeder Adapter läßt sich in vier logische Komponenten unterteilen (Abbildung 4 auf Seite 11):

1. Kabel-Interface
2. Puffer und Steuerungslogik
3. Mikroprozessor
4. Geräte-Interface

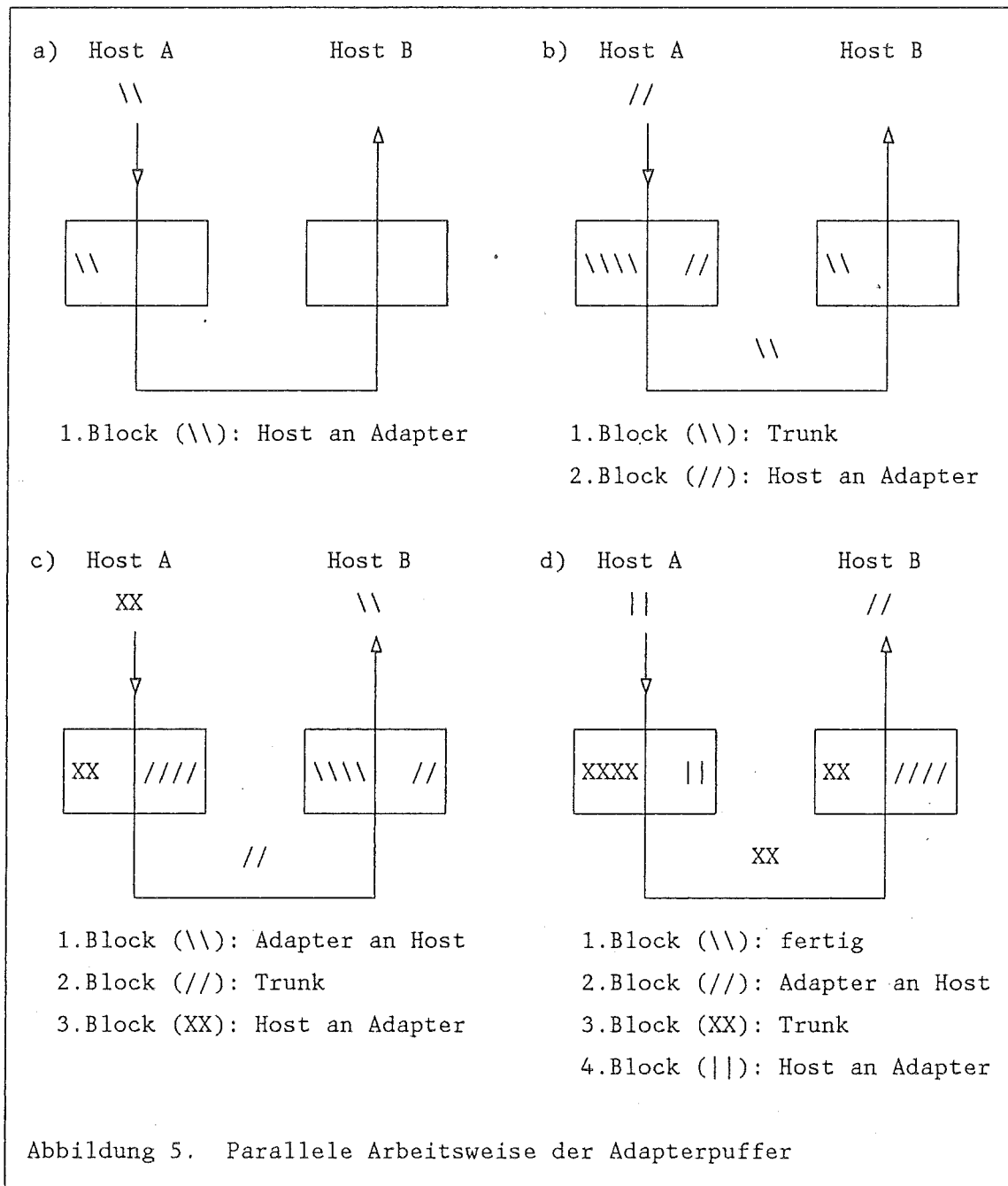


Während das Geräte-Interface von dem angeschlossenen Rechner bzw. Peripheriegerät abhängt, sind die anderen Komponenten in allen Adaptern identisch. Der Datenspeicher wurde als 'Wechselpuffer' organisiert, d.h. er enthält zwei logische Bereiche mit je 2 bzw. 4 Kbyte, die parallel gefüllt oder geleert werden können. Dies ermöglicht einen gleichzeitigen Datentransfer zu dem angeschlossenen Gerät oder zu dem Netz (Abbildung 5 auf Seite 13). Auf diese Weise wird erreicht, daß der Adapter nicht jedesmal auf das Ende des Paketes warten muß und er es erst danach weiterreichen

kann, sondern schon ein neues Paket ankommt, während er noch mit der Übertragung des alten Pakets beschäftigt ist.

Zu beachten ist, daß die hohe Übertragungsgeschwindigkeit des HYPERchannel/50 durch einen erheblichen Hard- und Softwareaufwand erkauft wird. Damit die Adapter in dem Netz keinen Engpaß erzeugen ist eine schnelle Treiberelektronik notwendig /NSC 85/:

- für die notwendige Logik wird ein Mikroprozessor mit einer Zykluszeit von 320 ns verwendet;
- als Speicher werden ECL-chips mit einer Zugriffszeit von 80 ns zum Puffern der Nachrichten benutzt;



3.1.2 Protokoll

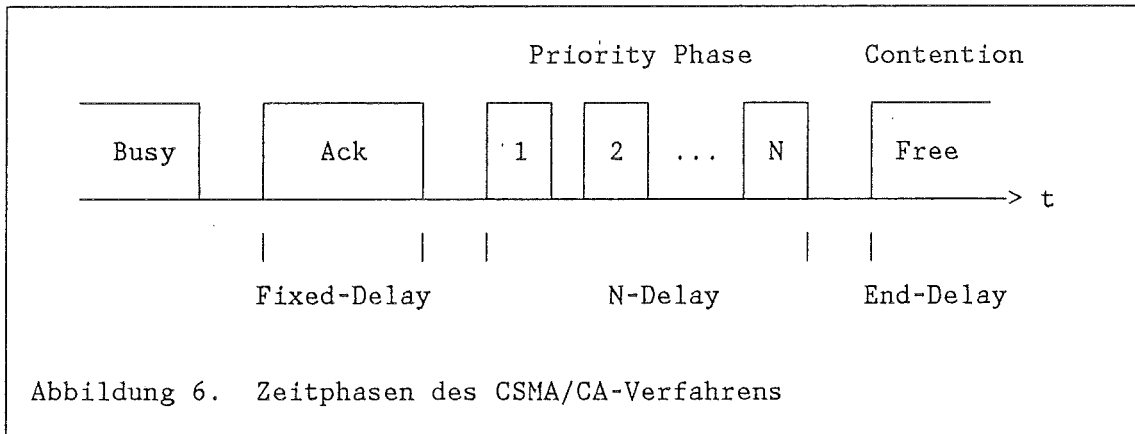
Es war das Ziel ein stabiles Protokoll zu entwickeln, welches mit wachsender Last auch steigenden Durchsatz garantiert. Der HYPERchannel/50 verwendet ein 'Carrier Sense Multiple Access with Collision Avoidance'

Protokoll. Bei diesem CSMA/CA Verfahren handelt es sich um ein modifiziertes CSMA/CD Protokoll /Kauf 84/.

Der wesentliche Unterschied zu dem von Ethernet benutzten 1-persistent CSMA/CD liegt in dem geänderten Wiederaufsetzverfahren nach einer Kollision /Ethe 80/. Bei dem Ethernet-Verfahren hört jede sendewillige Station ununterbrochen den Kanal ab, und sendet unverzüglich, nachdem er frei geworden ist. Ein Konflikt entsteht mit Sicherheit, wenn mehrere Stationen auf das Ende einer Übertragung warten. Daraufhin kommt das Binary-Exponential-Backoff Schema zum Zug und es wird durch die Wahl zufälliger Wartezeiten versucht diesen Zusammenstoß zu lösen. Da jedoch weiterhin kein spezielles Verfahren zur Lösung eines solchen Konfliktes existiert, kann es zu immer neuen Kollisionen kommen. Dies führt bei mittlerer bis hoher Last zu sehr großen Wartezeiten und einer erheblichen Performance-Verschlechterung /Ethe 80/.

Das HYPERchannel/50-Protokoll hingegen verwendet feste Adapter-Prioritäten. Die Intention dieser Designmodifikation war /Fran 84/:

- sicherzustellen, daß sendewillige Adapter Rücksicht auf laufende Übertragungen nehmen;
- Acknowledgements mit der höchsten Priorität empfangen und ohne Interferenz übertragen zu können;
- ein Prioritätenschema zu unterstützen, welches es der Station mit der höchsten Priorität erlaubt, so viel wie gewünscht zu übertragen, und der zweithöchsten Station erlaubt, so viel wie gewünscht von der verbleibenden Kapazität zu nutzen usw.;
- Konflikte zu lösen ohne zufällige Wiederaufsetzzeitpunkte zu generieren, so daß sichergestellt wird, daß kollidierte Nachrichten nun ohne erneuten Zusammenstoß übertragen werden können;



Dieses Zugriffsverfahren wird mit Hilfe von Timern, die in jedem Adapter installiert sind, realisiert. Ein Timer wird initialisiert und zurückgesetzt, sobald die Leitung busy ist, und er wird wieder freigegeben nachdem die Leitung idle ist. Nun zählt der Timer hoch und signalisiert nacheinander die einzelnen Zeitphasen (Abbildung 6).

Das HYPERchannel/50 Zugriffsprotokoll verwirklicht diese Phasen durch folgenden Mechanismus /Fran 84/ /Nsc 85/.

1. **Transmitter disable:** Wenn festgestellt wird, daß sich eine Message auf der Leitung befindet (Busy-Phase), sind alle Adapter bis auf den Sender blockiert. Auf diese Weise wird sichergestellt, daß laufende Übertragungen nicht durch andere Adapter gestört werden können. Diese Rücksichtnahme gilt auch dann, wenn die an der momentan stattfindenden Sendung nicht beteiligten Adapter eigentlich sendeberechtigt wären, d.h. sie sich in ihrem 'window' (N-Delay) befinden.
2. **Fixed - Delay:** Während einer festen Verzögerungsphase, die jeder Übertragung folgt, darf nur der empfangende Adapter senden. Diese Phase erlaubt dem Empfänger unmittelbar nach Erhalt der Sendung, eine Quittung an den Sender zu schicken, ohne mit einer Kollision rechnen zu müssen. Falls eine Quittung erwartet, aber nicht während des Fixed-Delays empfangen wird, nimmt der sendende Adapter an, daß die Nachricht verloren gegangen ist und bemüht sich um eine erneute Übertragung. Um sicherzustellen, daß während der Wartezeit auf die Antwort kein anderer Adapter eine Sendung beginnt, darf das Fixed-Delay erst enden, wenn die Responsemessage alle Stationen erreicht hat. Das

Fixed-Delay (fd) ist somit für alle Adapter gleich lang und hängt von der Gesamtlänge des Trunks, angegeben in Meter (d_{\max}), ab:

$$fd = 2 * pt * d_{\max} + 2080 \text{ ns}$$

wobei pt die Ausbreitungsgeschwindigkeit (Propagation Time) des elektrischen Impulses auf dem Medium darstellt. Diese Geschwindigkeit wird in hochwertigen Koaxialkabeln mit 78 % der Lichtgeschwindigkeit angegeben. Daraus ergibt sich für pt ein Wert von 4 ns/m. Bis auf den additiven Korrekturwert von 2080 ns, welcher der Zeit, die zur Formulierung und dem Beginn der Übertragung eines Antwort-Frames notwendig ist, entspricht, stellt das Fixed-Delay zweimal die End-to-End Propagation Time des Trunks dar.

3. N - Delay: Der Fixed-Delay Periode folgt die Prioritätenphase. Die Übertragungen während dieser Phase werden durch die Prioritätenreihenfolge der Adapter bestimmt. Jedem Adapter ist eine eindeutige Verzögerungszeit in Abhängigkeit seiner Priorität und seiner Position auf der Leitung zugewiesen. Eine sendewillige Station kann nur während ihres N-Delays eine Übertragung starten und hat dafür jedoch die Gewähr, daß ihre Nachricht nicht kollidiert. Wenn ein Adapter sein N-Delay verpasst hat, muß er entweder auf den nächsten für ihn vorgesehenen Abschnitt oder auf ein End-Delay warten. Je höher die Adapterpriorität ist, desto kürzer ist sein N-Delay. Für jeden Adapter muß das N-Delay so gewählt werden, daß seine Sendungen nicht mit den Sendungen des Adapters mit der nächst höheren Priorität interferieren können. Die Verzögerung des ersten Adapters wird durch die Zeit, welche zur erneuten Präparation einer nicht quittierten Sendung notwendig ist, bestimmt. Weiterhin ergibt sich für die Adapter k (mit $k=2, \dots, n$):

$$nd_1 = fd + 480 \text{ ns} \quad (\text{N-Delay für Adapter 1})$$

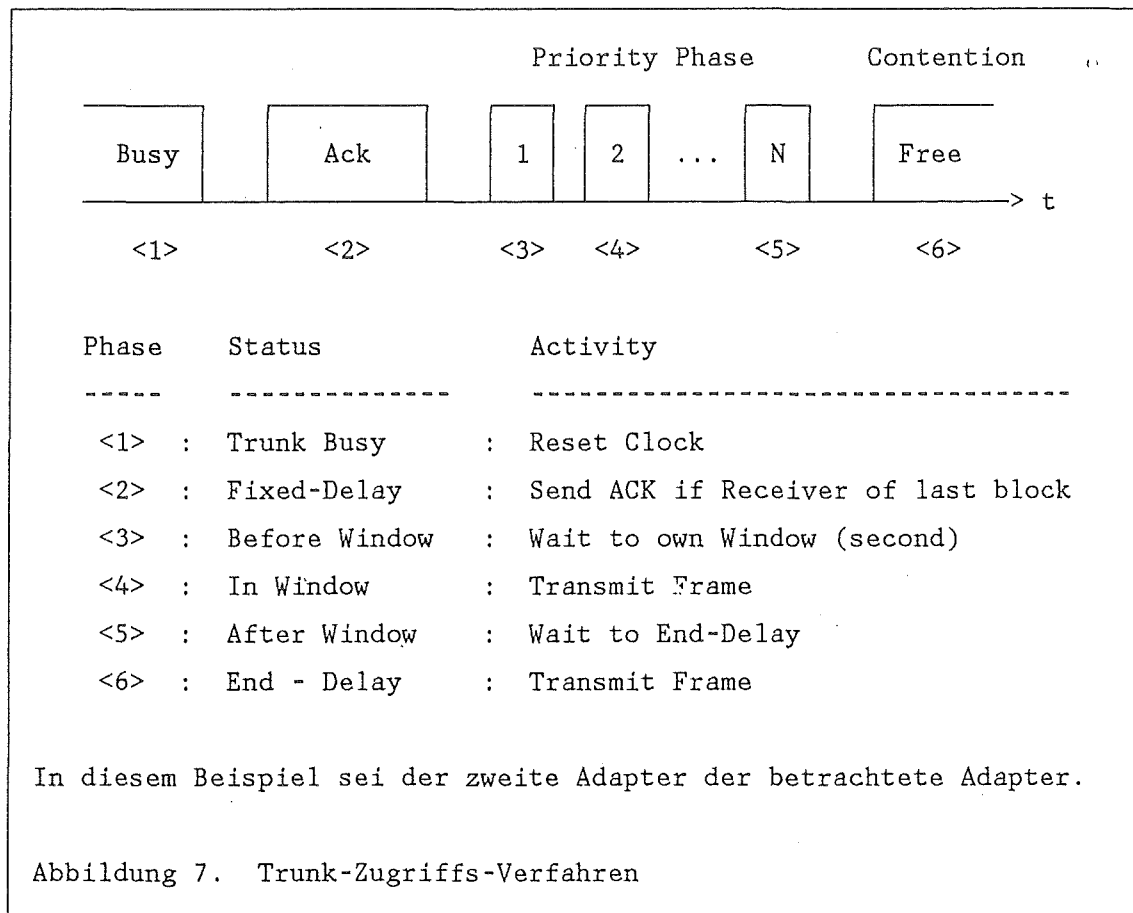
$$nd_k = nd_{k-1} + 2 * pt * d_{k-1,k} + 1600 \text{ ns} \quad (\text{N-Delay für Adapter k})$$

Dabei beschreibt $d_{k-1,k}$ die Entfernung vom Adapter k-1 bis zum Adapter k und der Wert von 1600 ns wird hinzugefügt, um durch Leitungsreflexionen entstehende Probleme zu vermeiden.

4. **End - Delay:** Falls eine komplette N-Delay Sequenz verstreicht, ohne daß ein Adapter etwas gesendet hat, tritt ein End-Delay ein und die Konkurrenzperiode beginnt. In dieser Phase werden alle Adapter freigegeben (enabled) und jeder darf übertragen. Wenn zwei oder mehr Adapter während dieser Periode die Leitung für frei befinden und fast gleichzeitig eine Sendung starten, kann es zu Zusammenstößen kommen. Die Länge des End-Delay wird durch das N-Delay des Adapters mit der niedrigsten Priorität plus einem Zuschlag, um die letzte Sendung sicher passieren zu lassen, bestimmt. Es ergibt sich für den Adapter k (mit $k=1, \dots, n$):

$$ed_k = nd_n + 2 * pt * d_{k,f(k)} + 1600 \text{ ns}$$

Hierbei ist zu beachten, daß nd_n das N-Delay des Adapters mit der niedrigsten Priorität darstellt und, daß es sich bei $f(k)$ um eine Funktion handelt, welche den am weitesten vom Adapter k liegenden Adapter ermittelt. Damit liefert $d_{k,f(k)}$ die größte Entfernung von k zu allen anderen Adaptern auf dem Trunk.



In Abbildung 7 sind die sechs auftretenden Phasen und die entsprechenden Maßnahmen der Adapter aufgeführt.

In Unterschied zum Ethernet Protokoll wird durch das CSMA/CA Verfahren erreicht, daß pro übertragener Nachricht höchstens **ein** Konflikt entsteht. Insgesamt ergibt sich dadurch ein viel angenehmeres Hochlastverhalten.

3.1.2.1 Frames und Framesequenzen

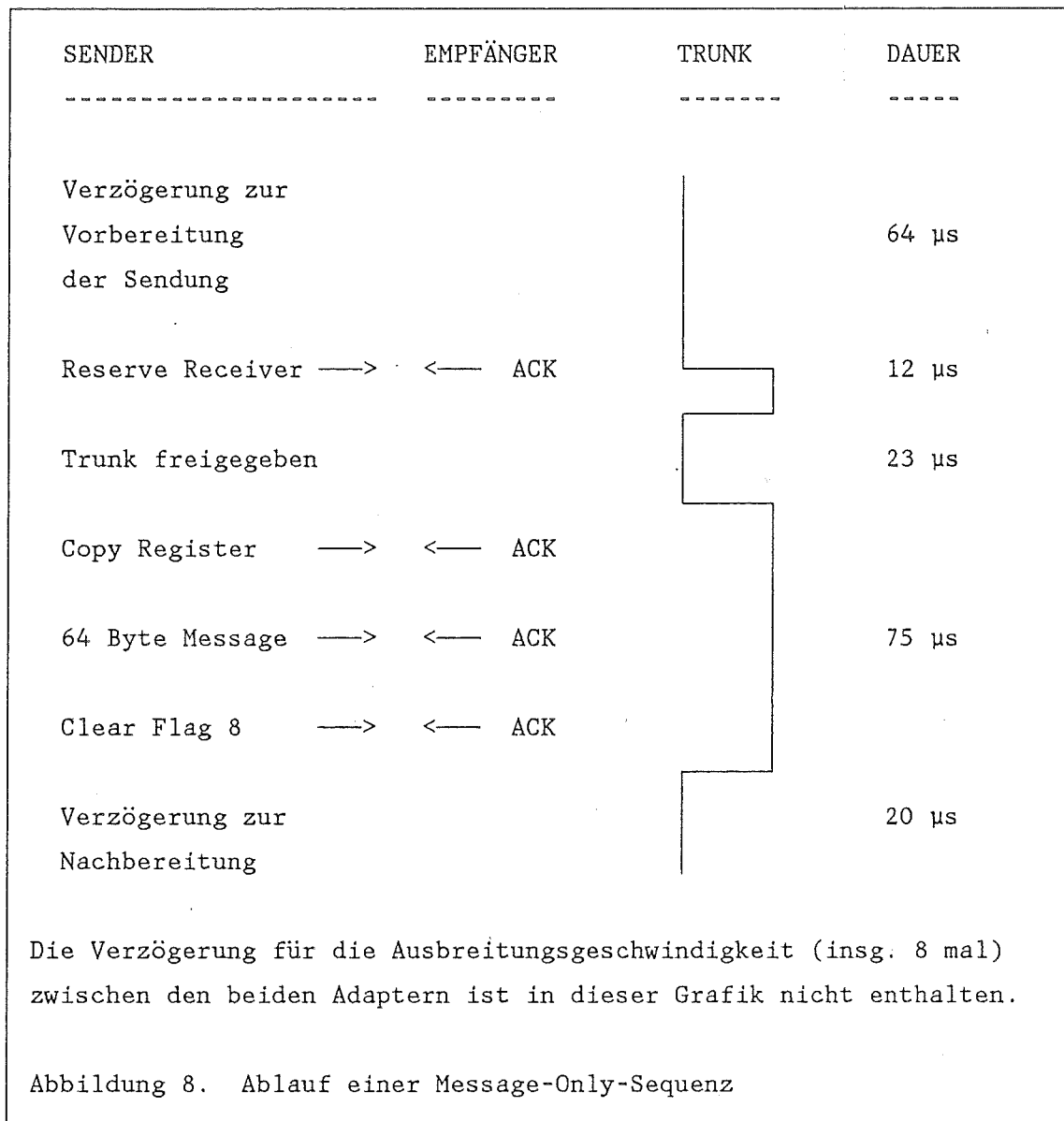
Das HYPERchannel/50-Data-Link-Protokoll läßt sich in zwei Informations-ebenen untergliedern. Das Element der tieferen Protokollebene ist ein Frame und die Struktur der höheren Ebene ist eine Framesequenz /Fran 82/.

Die Frames, welche die kleinste übertragbare Informationseinheit darstellen, lassen sich in drei unterschiedliche Typen einteilen:

1. Transmission-Frames sind sehr kurz (bis zu 38 Bytes) und werden benutzt, um Signale und Statusberichte zwischen den Adaptern auszutauschen (Abbildung 9 auf Seite 22).
2. Data-Frames treten in zwei Formen auf:
 - a. Sogenannte Message-Propor-Frames werden benutzt um kurze Nachrichten zu schicken;
 - b. Größere Blöcke werden mittels Data-Frames, die unmittelbar den vorrausgeschickten Message-Propor-Frames folgen, übertragen;
3. Einer erfolgreichen Übertragung einer der obigen Frames folgt immer ein Response-Frame. Es dient als Quittung für den Empfang einer Sendung.

Die Framesequenz als Struktur der höheren Protokollebene erhält man, wenn mehrere einzelne Frames zu einer Gruppe zusammengefaßt werden. Zwischen zwei verschiedenen Gruppen läßt sich differenzieren:

- Eine Message-Only-Sequenz erlaubt die Übertragung eines Message-Propor und beinhaltet bis zu 64 Bytes an Informationen. Diese Sequenz wird für kurze Sendungen verwendet.
- Größere Datenbestände werden in 2 Kbyte (optional 4 Kbyte) Blöcken als Message-With-Data-Sequenzen gesendet.



Um eine effiziente Ausnutzung des Trunks zu erreichen, behält der sendende Adapter nicht während der gesamten Zeit einer Sequenzübertragung die Kontrolle über die Leitung. Vielmehr gibt er den Trunk frei, solange er eigene Arbeiten ausführt und ihn nicht benötigt (Abbildung 8).

Nach der Übermittlung des ersten Frames einer Sequenz wird der Trunk freigegeben. Sobald der Adapter bereit ist den Rest der Sequenz zu übertragen, muß er sich erneut, in Konkurrenz zu den anderen Adaptern, um die Leitung bewerben. Nachdem er die Kontrolle über den Trunk zurück erhalten hat, sendet er ein kurzes Transmission Frame (Copy Register) an den empfangenden Adapter. Falls dieses Copy Register Frame ohne Kollision ankommt,

schickt der Empfänger ein Response Frame. Nachdem der Sender die Quittung empfangen hat, schickt er während der Fixed-Delay Periode das nächste Frame aus seiner Sequenz. Da alle anderen Adapter Rücksicht auf die laufende Übertragung nehmen müssen, geschieht dies ohne Zusammenstöße mit anderen Sendungen.

In dieser Phase wird deutlich, daß das Fixed-Delay nicht nur zur Quittierung vorangegangener Messages benutzt wird, sondern auch um eine komplette Sequenz von Nachrichten kollisionsfrei zu übertragen. Zwar werden mit Beginn des Fixed-Delays die Timer in allen Adaptern neu gestartet, jedoch können die nicht an der laufenden Sequenz beteiligten Adapter ihr Übertragungsrecht während ihres N-Delays nicht wahrnehmen, da bereits eine im Laufe des Fixed-Delays begonnene Sendung stattfindet.

Falls es sich bei der laufenden Sequenz um eine Message-Only-Sequenz handelt, wird die Kontrolle über den Trunk erst nach Übertragung des Restes der Sequenz zurückgegeben.

Set Reserve	Dies ist der Wunsch den empfangenden Adapter zu reservieren.
Copy-Register	Die Aufforderung an den empfangenden Adapter Statusinformationen zu übermitteln. Dieses Frame wird auch zu Beginn eines Message- bzw. Datenblocks geschickt, um den Trunk wieder zu erlangen.
Flag-8	Das Flag 8 wird in dem empfangenden Adapter gesetzt, wenn dieser ein Message-Propor Frame erwartet. Das Clear-Flag-8 löscht Flag 8, sobald die Nachricht angekommen ist.
Flag-9	Der Empfänger setzt Flag 9, wenn er bereit ist einen (weiteren) Datenblock zu empfangen. Es wird vom Sender im empfangenden Adapter zurück gesetzt nachdem ein Datenblock empfangen wurde. Falls es im sendenden Adapter gesetzt ist, so bedeutet dies, daß er bereit ist Daten zu übertragen. Das Flag-9 muß vom Empfänger im Sender gelöscht werden, wenn der Empfänger in der Lage ist Daten aufzunehmen.
Flag-A	Dieses Flag wird gesetzt, wenn ein Datenblock folgt. Es wird erst zurückgesetzt nachdem alle Datenblöcke dieser Sequenz erfolgreich übertragen wurden, und die virtuelle Verbindung zwischen Sender und Empfänger terminiert werden kann.

Abbildung 9. Transmisson Frames

Bei einer Message-Data-Sequenz wird nach Übertragung des Message Proper die Leitung freigegeben. Der Empfänger weiß, daß er einen Datenblock aufnehmen soll und bereitet dies vor. Sobald er den dafür notwendigen Speicher im Adapterpuffer zur Verfügung hat, bewirbt er sich um den Trunk. Nachdem er das Zugriffsrecht erhalten hat, schickt er dem Sender ein Clear-Flag-9 Frame, um ihm seine Bereitschaft zur Datenaufnahme mitzuteilen. Daraufhin erfolgt ein Austausch von Statusinformationen mittels des Copy-Register-Frames. Nun kann die eigentliche Übertragung der Daten stattfinden, die von dem Sender mit einem Clear-Flag-9 Frame abgeschlossen wird. Dieser Vorgang wird solange für jeden Block von 'Associated-Data'

wiederholt, bis die gesamte Datei übertragen wurde. Dabei muß sich der Empfänger nur zu Beginn einer jeden Teilsequenz um die Leitung bewerben. Während der folgenden Phase dieser Teilsequenz wird der Trunk nicht freigegeben, womit erreicht wird, daß niemals ein Datenpaket in eine Kollision verwickelt ist. Vielmehr beschränken sich die möglichen Zusammenstöße auf das Clear-Flag-9 Frame, welches relativ kurz ist und ohne viel Aufwand neu gesendet werden kann. Die Reservierung des Trunks erfolgt durch die Übertragung von Synchronisationsbits, solange der Adapter mit der Vorbereitung des nächsten Frames beschäftigt ist. Da die Übertragung während des Fixed-Delay beginnt und über die gesamte Präparierungszeit andauert, bleibt der Trunk im Besitz des Senders.

Alternativ zu diesem Schema existiert ein 'Burst-Mode'-Verfahren, welches sogar die Zeiten zwischen den einzelnen Teilsequenzen der Datenblöcke überbrückt und so sicherstellt, daß ein ganzes File übertragen werden kann, ohne den Trunk zwischendurch freizugeben. Bei diesem Verfahren wird die Leitung jedoch sehr lange von einem Adapter belegt, und die Präparierungszeiten dieses Adapters können nicht sinnvoll von anderen Teilnehmern zur Datenübertragung genutzt werden. Außerdem werden bei Übermittlung von großen Dateien die Wartezeiten für die übrigen Benutzer unzumutbar lang. Aus diesen Gründen wird von dem 'Burst-Mode'-Verfahren in der Praxis auch kaum Gebrauch gemacht /Fran 84/.

TEILAKTION innerhalb der Message-Only-Sequenz	ZEIT	D E L A Y
-----	----	-----
Übertragung des Set-Reserve-Frames	5.2	
Ausbreitungsverzögerung		prop-time
Zeitverzögerung um ACK zu erstellen	2.8	(*)
Übertragung des ACK-Frames	3.7	
Ausbreitungsverzögerung		prop-time
Freigabe des Trunks	23.0	
Übertragung des Copy-Register-Frames	5.2	
Ausbreitungsverzögerung		prop-time
Zeitverzögerung um ACK zu erstellen	2.8	(*)
Übertragung des ACK-Frames und der Register	6.65	
Ausbreitungsverzögerung		prop-time
Ausführung des Mikrocodes, Trunk bleibt belegt	17.0	(*)
Übertragung des 64-Byte-Message-Frames	13.6	
Ausbreitungsverzögerung		prop-time
Zeitverzögerung um ACK zu erstellen	2.8	(*)
Übertragung des ACK-Frames	3.7	
Ausbreitungsverzögerung		prop-time
Ausführung des Mikrocodes, Trunk bleibt belegt	10.0	(*)
Übertragung des Clear-8-Frames	3.1	
Ausbreitungsverzögerung		prop-time
Zeitverzögerung um ACK zu erstellen	2.8	(*)
Übertragung des ACK-Frames	3.7	
Ausbreitungsverzögerung		prop-time
Hardwareverzögerung	3.2	
<p>Bei den mit (*) gekennzeichneten Aktionen handelt es sich um Software Delays. Die Zeiten sind in Mikrosekunden angegeben. Für die Variable 'prop-time' muß die zeitliche Verzögerung durch die Ausbreitungsgeschwindigkeit des Signals zwischen den betroffenen Adaptern eingesetzt werden.</p>		
<p>Abbildung 10. Zeiten für eine Message-Only-Framesequenz</p>		

3.1.2.2 Wait-Flip-Flop

Bei dem oben beschriebenen HYPERchannel/50-Zugriffsverfahren kann es vorkommen, daß der Channel von dem Adapter mit der höchsten Priorität monopolisiert wird. Dies ist möglich, da nach jeder Übertragung ein Fixed-Delay und die N-Delay Phase folgt. Während des ersten N-Delays, welches für den Adapter mit der höchsten Priorität vorgesehen ist, kann dieser ohne mit Schwierigkeiten rechnen zu müssen mit einer neuen Sendung beginnen. Alle folgenden Adapter müssen Rücksicht auf die laufende Übertragung nehmen und ihr N-Delay verstreicht, ohne daß sie es wahrnehmen konnten. Um dies zu verhindern, ist jeder Adapter mit einem Wait-Flip-Flop (WFF) ausgestattet, welches gesetzt wird, wenn der Adapter überträgt, und wieder zurückgesetzt wird, wenn die Konkurrenzphase - signalisiert durch das End-Delay - beginnt. Solange das WFF gesetzt ist, kann der Adapter keine Übertragung starten. Auf diese Weise soll ein faires Round-Robin ähnliches Zugriffsverfahren erreicht werden. Das WFF muß jedoch nicht notwendigerweise aktiviert sein und es können sogar, da jeder Adapter mit einem WFF ausgestattet ist, gleichzeitig auf einem Trunk sowohl Adapter mit aktiviertem WFF als auch Adapter mit deaktiviertem WFF koexistieren.

3.1.2.3 Virtuelle Adapter-Verbindungen

Die Schaffung einer virtuellen Verbindung zwischen zwei kommunizierenden Adaptern während der Dauer einer Framesequenz wird durch das Adapterreservierungsschema erreicht. Dieses Schema läßt sich in fünf Phasen unterteilen:

1. Nachdem ein Adapter eine Message vom Host empfangen hat, welche übertragen werden soll, reserviert er sich selbst. Dies bedeutet, daß der Adapter nur noch Sendungen vom dem im Übertragungswunsch des Host festgelegten Empfänger annimmt.
2. Sobald der Adapter sich selbst reserviert hat, versucht er den Empfänger zu reservieren, indem er ihm ein Set-Reserve-Frame schickt.

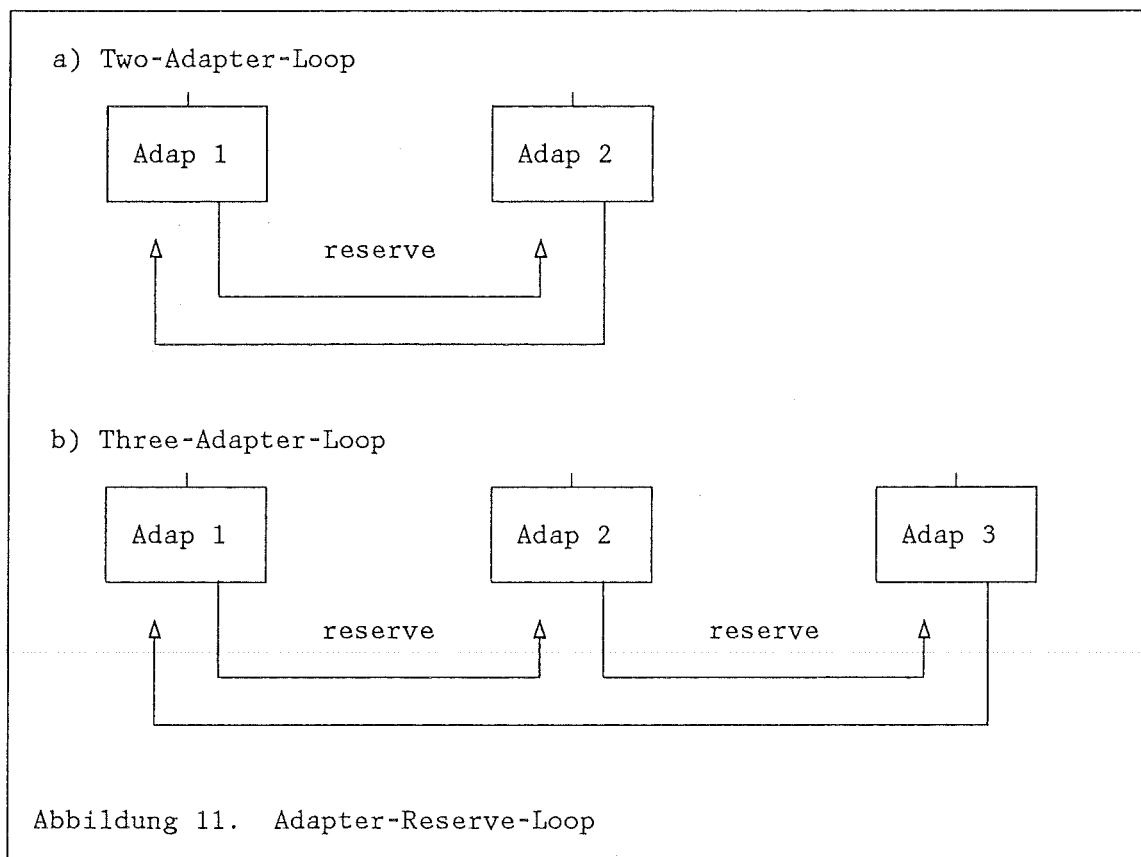
3. Falls der Empfänger frei ist, d.h. er hat weder sich selbst reserviert, noch wurde er von einem anderen Teilnehmer belegt, geht er in den Zustand Reserved über. Nun informiert er den Sender, daß der Reservierungsversuch erfolgreich war und bestätigt damit den Aufbau der virtuellen Verbindung. Daraufhin fährt die Übertragung der Framesequenz wie unter 3.1.2.1. beschrieben fort. Der sendende wie auch der empfangende Adapter bleiben während der gesamten Sequenz reserviert.
4. Nach der Übertragung des Message-Header und aller zugehörigen Daten, läßt der Sender den Empfänger mittels eines End-Of-Transmission Frames (Clear Flag A) frei.
5. Zum Schluß gibt der Sender sich selbst frei und beendet damit die virtuelle Verbindung.

Diese Phasen laufen in der Realität nicht immer wie hier beschrieben hintereinander ab. Es müssen somit Vorkehrungen getroffen werden, um Fehler innerhalb des Auf- und des Abbaus einer virtuellen Verbindung zu vermeiden. In Phase 2 ist es möglich, daß der Sender einen schon reservierten Adapter als Empfänger belegen möchte. In diesem Fall lehnt der Empfänger die Reservierung ab und schickt ein Reservation-Reject Frame an den Sender. Die bereits bestehende Verbindung des Empfängers mit seinem Sender wird dadurch nicht beeinflusst. Der Adapter, welcher den erfolglosen Reservierungsversuch durchgeführt hat, möchte jedoch weiterhin mit dem belegten Empfänger eine Verbindung aufbauen. Dafür soll er einerseits möglichst schnell den Empfänger belegen, nachdem dieser frei geworden ist, andererseits darf er den Trunk nicht permanent mit seinen Reservierungsversuchen belasten. Der im HYPERchannel/50-Protokoll realisierte Ansatz, um dieses Dilemma zu lösen benutzt ein 'Binary-Exponential-Time' Verfahren. Dieses Verfahren ist nicht mit dem aus dem Ethernet bekannten 'Binary Exponential Backoff' identisch, obwohl es Ähnlichkeiten aufweist:

Nach dem ersten erfolglosen Reservierungsversuch läßt der Sender eine Zeitverzögerung von einer Mikrosekunde verstreichen, bevor er einen zweiten Versuch unternimmt. Falls er feststellt, daß sein Bemühen nicht zum gewünschten Resultat führte, verdoppelt er nun bei jedem weiteren Versuch seine Zeitverzögerung, bis die Verzögerung 128 ms beträgt. Sollten

dann die bisherigen 8 Versuche nicht zur Belegung des Empfängers geführt haben, so wird seine Verzögerungszeit wieder mit einer Mikrosekunde initialisiert und eine neue Delay-Sequenz durchlaufen. Der Adapter fährt solange mit seinen Versuchen fort bis er erfolgreich ist, oder bis die durchgeführte Anzahl von Delay-Sequenzen eine als 'Retry Counter' vorgegebene Zahl überschreitet. Wenn dieser Fall eintritt blieb der Sender erfolglos bei seinen Versuchen einen anderen Adapter als Empfänger zu reservieren und seine Transmission wird abgebrochen, worauf der Sender in den Zustand 'idle' zurück kehrt.

Während der gesamten Phase der Reservierungsbemühungen bleibt der Sender durch sich selbst belegt und somit für andere Adapter nicht verfügbar. Die 'Binary Exponential' Verzögerung zwischen den Reservierungsversuchen dient als 'Flow Control' Mechanismus, welche verhindert, daß der sendende Adapter unnützerweise den Trunk monopolisiert. Der 'Retry Counter' eines jeden Adapters kann individuell gewählt werden und bewegt sich im Bereich von 55 bis 250.



Das Adapterreservierungsschema wie oben beschrieben birgt die Gefahr eines Adapter-Deadlocks. Falls zwei Adapter quasi gleichzeitig versuchen sich gegenseitig zu reservieren, findet jeder Adapter den anderen schon belegt vor und beginnt mit seinem 'Binary Exponential Time' Verfahren (Abbildung 11). Da jedoch keine der beiden Stationen erfolgreich ist in ihrem Bemühen den anderen zu reservieren, fahren Sie solange fort, bis der Retry Counter eines der beiden Adapter abgelaufen ist. Um dieses Gefahrenpotential zu mindern erhält ein Sender, welcher keinen Erfolg bei der Reservierung seines erhofften Empfängers hatte, zusammen mit dem Not-Acknowledge die Nummer desjenigen Adapters, welcher der Empfänger versucht zu reservieren. Wenn ein Adapter nach Empfang der ablehnenden Antwort feststellt, daß er und der ablehnende Adapter versuchen sich gegenseitig zu reservieren, bricht er seinen Übertragungswunsch ab und gibt sich frei. Auf diese Weise versetzt er sich in die Lage den Reservierungswunsch des anderen Adapters entgegen zu nehmen und zu akzeptieren.

Die 'Trägheit' des Systems, d.h. die Verzögerungen durch die Ausbreitungsgeschwindigkeit, die Elektronik etc., kann dazu führen, daß beide Adapter ihre Versuche den anderen zu reservieren abbrechen und somit keiner von beiden in seiner Übertragung fortfahren kann /Donn 79/. Die Zeit, die benötigt wird um nach dem Empfang einer ablehnenden Antwort zu erkennen, daß es sich um einen gegenseitigen Reservierungsversuch handelt, bis zur Freigabe des eigenen Adapters beträgt ungefähr 50 Mikrosekunden.

Außer der oben erwähnten Deadlock Gefahr existiert eine weitere Deadlock Möglichkeit, welche von dem Backoff Mechanismus nicht gelöst werden kann. Dabei handelt es sich um den Fall, daß drei oder mehr Adapter fast gleichzeitig versuchen einer den anderen zu belegen, d.h. sich in einem 'Reservation Request Loop' befinden. Dies kann eintreten, falls Adapter A versucht Adapter B zu reservieren, während B versucht C zu belegen und C wiederum Adapter A reservieren möchte; insgesamt also eine Reservierungsschleife entsteht. In einer solchen Situation haben die Adapter keine Möglichkeit zu erkennen, daß sie sich in einem Loop befinden und fahren deshalb mit ihrer erwähnten Retry-And-Delay Sequenz fort. Die Lage bessert sich erst, wenn der Retry Counter eines dieser Adapter abgelaufen ist und damit seine Reservierungsversuche enden und er sich selbst freigibt. Falls die anderen an dem Loop beteiligten Adapter nicht gleichzeitig ihre Versuche abbrechen, ist es sehr wahrscheinlich, daß einer der verbleibenden

Adapter beim nächsten Versuch erfolgreich ist und damit den Deadlock auflöst. Aus diesem Grund sollte sichergestellt sein, daß die an einem Loop beteiligten Adapter nicht zur selben Zeit ihre Versuche abbrechen. Das kann erreicht werden, indem der Retry Counter jedes Adapters in Abhängigkeit zu seiner Position im Netz gewählt wird und somit unterschiedlich für alle Adapter ist.

Das Adapterreservierungsschema kann in einem Netzwerk bei extrem hoher Last zu einer erheblichen Performance Verschlechterung führen /Donn 79/. Besonders bei einer sehr hohen Verkehrsdichte möchten alle oder fast alle Adapter eine Message übertragen und haben sich, in Übereinstimmung mit dem virtuellen Verbindungsprotokoll, selbst reserviert und versuchen nun ihren gewünschten Empfänger zu reservieren. Da jedoch alle oder fast alle sich selbst belegt haben, sind nur sehr wenige als Empfänger verfügbar, was zu einem temporären Deadlock führt. Dieser Deadlock wird aufgelöst, wenn, wie oben beschrieben, einer der an dem Loop beteiligten Adapter seinen Übertragungswunsch aufgibt und sich für eine Reservierung zur Verfügung stellt. Unter extrem hoher Last wird dieser Abbruch, trotzallem, nicht zu dem erhofften Ergebnis einer erfolgreichen Übertragung führen, da, bevor der aufgebende Adapter den Reservierungswunsch eines anderen Adapters akzeptieren kann, er von seinem Host einen neuen Übertragungswunsch erhält und sich aus diesem Grund wieder selbst reserviert, ohne vorher ein Request-Reserve Frame eines anderen Adapters akzeptiert zu haben.

Aus diesem Verhalten folgt, daß bei sehr hohem Verkehrsaufkommen die Adapter sich fast immer in einem 'Reservation Request Loop' Deadlock befinden, was als Resultat zu einer ernsten Verschlechterung des Durchsatzes führt. Zur Korrektur dieses Verhaltens innerhalb des virtuellen Verbindungsprotokolls wurden zwei Vorschläge gemacht /Fran 82/:

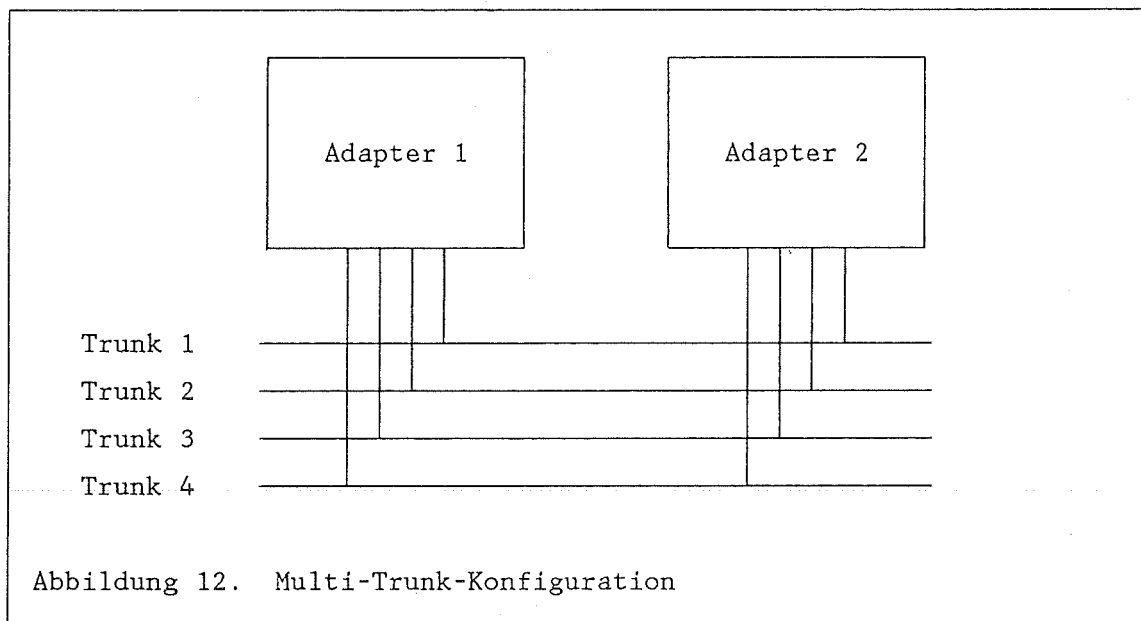
- Der erste Lösungsansatz, in /Donn 79/ zum ersten mal erwähnt, bezieht sich auf eine Modifikation des Reservierungsschemas, wobei eine lineare Ordnung, ähnlich der Prioritäten, unter den Adaptern erzeugt wird. Bei dem Aufbau einer virtuellen Verbindung zwischen zwei Adaptern wird der Adapter mit der höheren Nummer innerhalb dieser Ordnung zuerst reserviert, ungeachtet ob es sich dabei um den Sender oder den Empfänger handelt. Die Ergebnisse einer Simulation zeigten, daß eine solche Mo-

difikation zu einer substanziellen Verbesserung des Durchsatzes führten /Donn 79/.

- Eine alternative Lösung, vorgeschlagen in /Wats 79/, welche genauso effizient wie die Adapter Ordnung sein sollte, verlangte eine Änderung des 'Exponential Retry' Mechanismus. Es wurde empfohlen, daß die Adapter während ihrer Wiederholungsversuche nicht reserviert bleiben, sondern statt dessen sich jedesmal freigeben und so für andere Reservierungsversuche verfügbar sind.

3.1.2.4 Multi-Trunk Networks

In HYPERchannel/50 Netzwerken besteht die Möglichkeit bis zu vier Leitungen an jeden Adapter anzuschließen, d.h. man kann bis zu vier physische Verbindungen zwischen je zwei Adaptern aufbauen. Für jeden Trunk gilt das gleiche Adapterzugriffsprotokoll und jeder Trunk arbeitet unabhängig von den anderen.



Ein Adapter, welcher eine Nachricht senden möchte, muß sich um einen Trunk bewerben. Dazu untersucht er in einer vorgegebenen Reihenfolge alle ange-

geschlossenen Leitungen bis er einen freien Trunk gefunden hat. Trunks, die nicht probiert werden sollen, oder die nicht als physische Verbindung existieren, werden als 'disabled' bezeichnet. Trunks, die untersucht werden sollen, oder die eine Nachricht übertragen können, werden als 'enabled' bezeichnet. Ein Adapter probiert einen Trunk, indem er zuerst feststellt, ob der Trunk enabled ist. Falls dies der Fall ist, untersucht er ob der Trunk busy oder idle ist. Wenn der Trunk busy oder 'disabled' ist, probiert der Adapter den nächsten Trunk. Die Zeit, die benötigt wird um einen 'disable' Trunk zu probieren beträgt 2 Mikrosekunden, während die Untersuchung eines 'enable' Trunk 5.5 Mikrosekunden benötigt.

In der Message vom Host an seinen Adapter kann der Host dem Adapter mitteilen, welchen Trunk er benutzen soll bzw. welche Trunks er probieren soll. Die Leitungen, welche vom Empfänger zur Übertragung der Antwort-Frames benutzt werden sollen, können auch vom sendenden Host festgelegt werden. Falls in einem Netzwerk mehrere Trunks benutzt werden sollen, so ist man in der Art der Konfiguration nicht festgelegt. Es können sowohl alle Adapter miteinander über die gewünschte Anzahl von Leitungen verbunden werden (Fully Connected), als auch nur ausgewählte Adapter über eine hohe Anzahl von Verbindungen verfügen, während andere nur mit Hilfe eines Trunks zusammenhängen.

Auf diese Weise kann man nicht nur eine physische Zuordnung von Trunks und Adaptern erhalten, sondern auch eine logische Unterteilung von Nachrichten und Antworten auf verschiedene Trunks erreichen. Außerdem besteht die Möglichkeit große Pakete (Message-With-Data) auf einen anderen Trunk als kleine Pakete (Message-Only, Statusinformationen) zu schicken. Es ergeben sich also eine Vielzahl von Variationsmöglichkeiten.

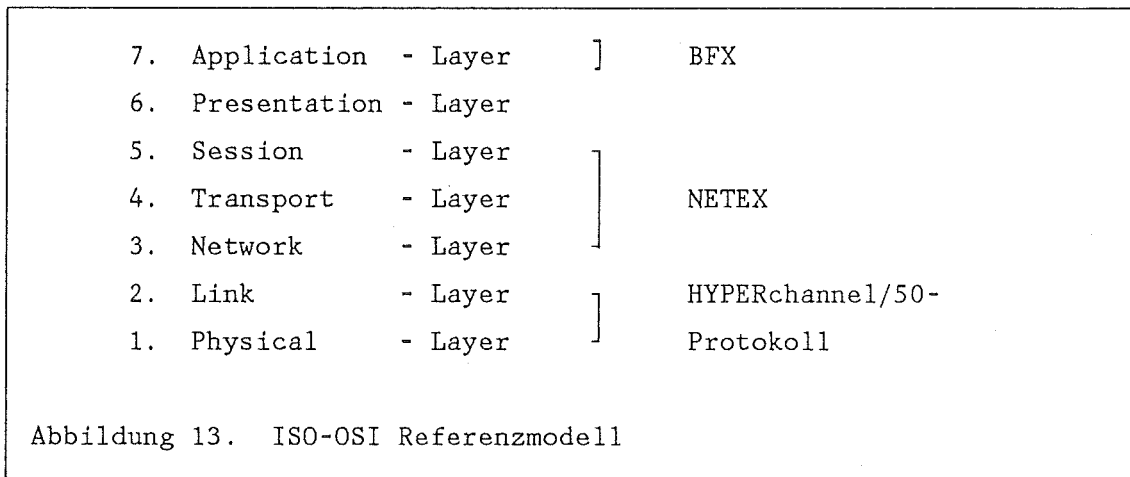
Es hat sich jedoch gezeigt, daß kein wesentlicher Vorteil aus der Verwendung mehrerer physischer Leitungen gezogen werden kann. Der Grund dieses zunächst unerwarteten Ergebnissen ist im Aufbau und der Speicherverwaltung der Adapter zu suchen. Zwar kann eine Nachricht vom Sender sofort über einen freien Trunk gesendet werden, aber ein Empfänger kann, wenn mehrere Nachrichten auf verschiedenen Trunks gleichzeitig bei ihm ankommen, nur eine von ihnen akzeptieren und schickt den anderen ein Reject-Frame. Der Speicher des Adapters wird für alle Trunks gemeinsam verwendet, d.h. es existiert kein Speicher pro Trunk, und er ist in seiner Größe ge-

nau für ein Datenpaket ausgelegt. Außerdem ist weder die Elektronik noch die Logik ('Buchführung' der ankommenden Pakete) in der Lage mehr als eine Nachricht zur gleichen Zeit zu verarbeiten. Trotzdem können sich bei Netzwerken, die nicht vollständig untereinander verbunden sind Durchsatzsteigerungen ergeben /Fran 84/.

Eine Unterteilung der Nachrichtentypen auf unterschiedliche Trunks führten, wie Simulationsstudien /Wats 79/ gezeigt haben, nicht zu den gewünschten Ergebnissen.

3.1.3 Software

Die HYPERchannel/50-Hardware deckt die beiden untersten Ebenen (Physical- und Linklayer) im ISO-OSI Referenzmodell ab. Für die darüber liegenden Ebenen 3 - 7 wird geeignete Software benötigt (Abbildung 13). Die Firma Network Systems Corporation stellt hierfür zwei Programmpakete zur Verfügung /NSC 85/:



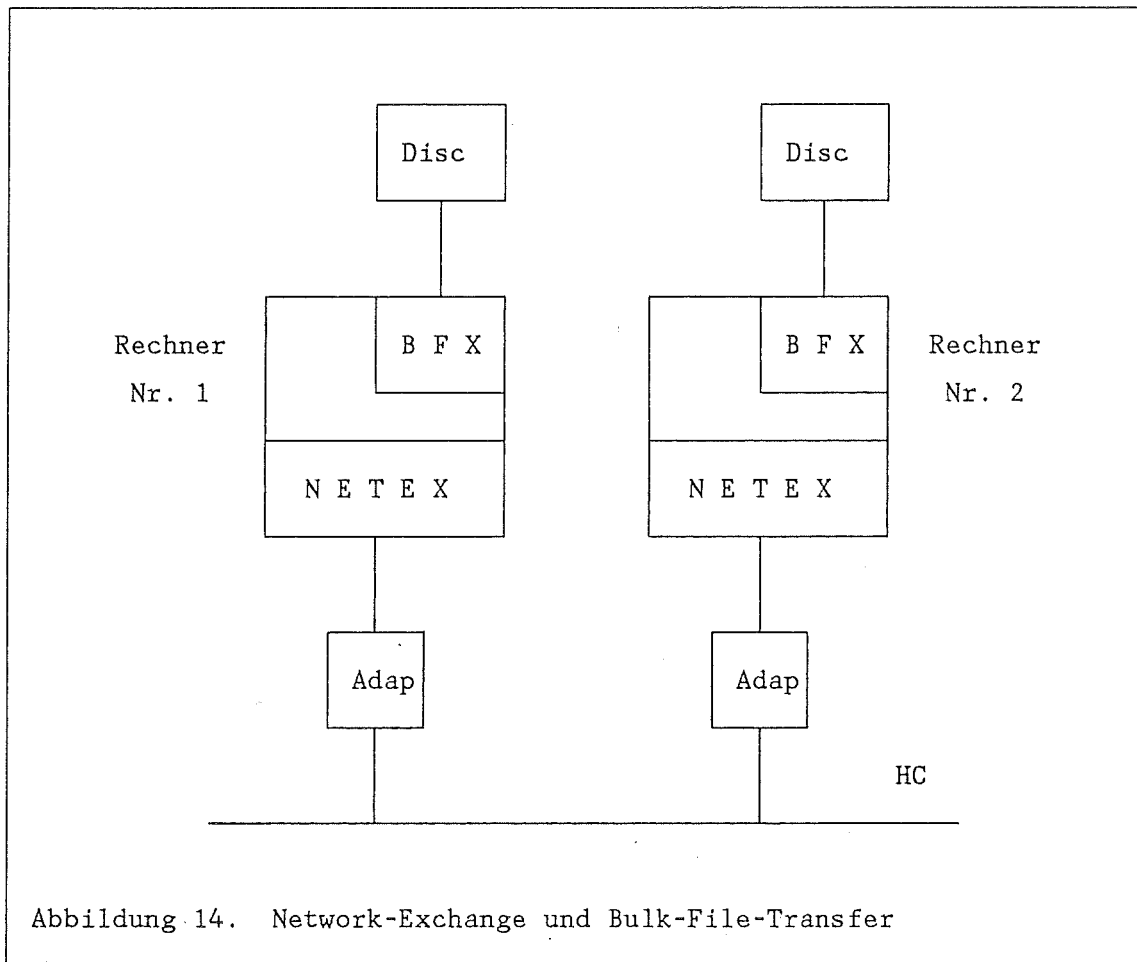
1. **NETEX** (Network Exchange): Dieses Program deckt in Anlehnung an das ISO-OSI Referenzmodell die Ebenen 3-5 ab (Network-, Transport- und Sessionlayer). Es stehen eine Reihe von Calls zur Verfügung, mittels derer eine Verbindung aufgebaut werden kann (Offer, Connect, Confirm), Blöcke transferiert werden können (Read, Write, Wait) und

die Session wieder beendet werden kann (Disconnect, Close). Die Organisation, d.h. die Buchführung, Adressverwaltung, Bereitstellung von Speicherplätzen, etc. wird von diesem Programm übernommen. Dazu muß es auf jedem Rechner, der am HYPERchannel/50 angeschlossen ist und der an dem Informationsaustausch teilnehmen will, installiert sein. Es handelt sich somit um eine Erweiterung des Betriebssystems in jeder einzelnen Anlage.

2. BFX (Bulk File Transfer): Als eine Anwendung in der Ebene 7 (Applicationlayer) erleichtert dieses Programmpaket die Übertragung großer Datenbestände. Zwischen den beiden Teilnehmern wird eine virtuelle Verbindung aufgebaut, um so ohne Unterbrechnug größere Dateien schicken zu können. (Abbildung 14 auf Seite 34). Dies geschieht durch die beiden Befehle:

- a. Transport initialising;
- b. Transport receiving;

mittels Job Submission, d.h. auf dem Rechner Nr. 1 steht eine Folge von Steuerbefehlen in der 'Job-Control-Language' des Rechners Nr. 2 zur Verfügung (Abbildung 14 auf Seite 34). Auf diese Weise können die Dateien von dem empfangenden Rechner sofort auf eine für ihn verständliche Weise bearbeitet werden. Dies schließt das Schreiben der Blöcke, das Führen der entsprechenden Kataloge, usw. ein.

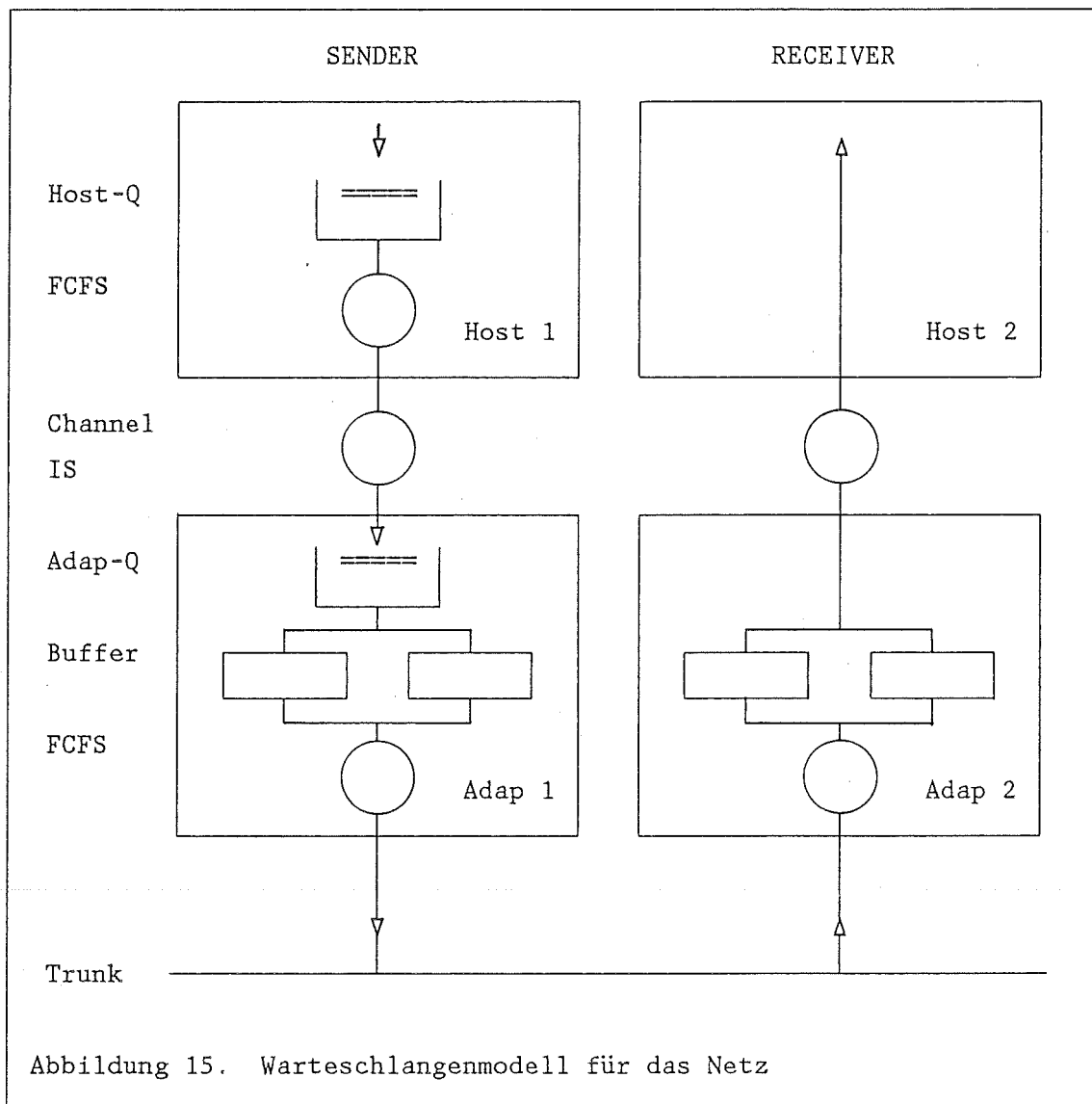


Die Möglichkeit der Modellbildung bietet den Vorzug, daß ein vereinfachtes Abbild der Wirklichkeit wiedergegeben werden kann. Mittels Abstraktion komplexer realer Zusammenhänge wird die Untersuchung großer und unübersichtlicher Systeme durchführbar.

Das Modell für den HYPERchannel/50 besteht, in Abhängigkeit der Verwendung des Adapters und des Hosts als Sender oder als Empfänger, aus unterschiedlichen Komponenten. Dieser differenzierte Aufbau des Warteschlangenmodells stellt sich wie folgt dar (Abbildung 15 auf Seite 36):

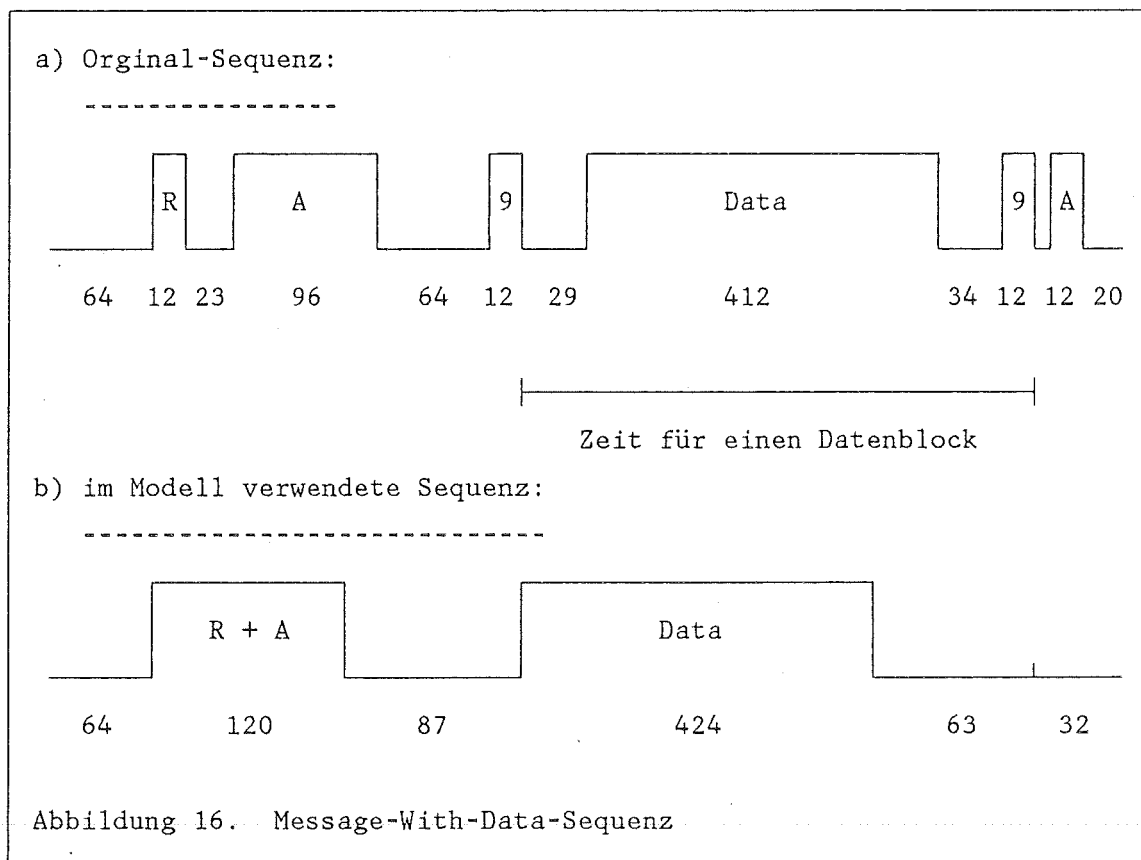
- Der sendende Host wird durch eine Warteschlange, welche die Übertragungswünsche aufnimmt, und eine Bedienstation, welche die Zeitverzögerung durch den Host widerspiegelt und als Zuteilungsstrategie das 'First-Come-First-Serve' Verfahren (FCFS) benutzt, abgebildet. An einen Kanal des Hosts wird der Adapter angeschlossen. Die Übertragungsrate des Kanals hängt von dem eingesetzten Host und dem verwendeten Kanaltyp (Blockmultiplex- oder Datastreamchannel) ab. Die zeitliche Verzögerung, welche die Daten während der Übertragung von dem Host zu dem Adapter erfahren, wird durch einen 'Infinite Server' (IS) dargestellt. Der sendende Adapter besteht, ähnlich wie der sendende Host, aus einer Warteschlange für die über den Kanal ankommenden Datenpakete und einer Bedienstation (FCFS) für die Adapteraktivität. Zwischen Warteschlange und Bedienstation befindet sich jedoch ein Pufferspeicher, der aus zwei unabhängigen Bereichen besteht und jeweils ein Paket aufnehmen kann. Im Unterschied zu einer Warteschlange kann sich immer nur ein Block in einem Pufferbereich befinden, d.h. jeder Bereich des Pufferspeichers ist entweder leer oder mit einem Paket gefüllt (siehe Kapitel "Hardware" auf Seite 10).
- Ist der Adapter und damit sein Host als Empfänger einer Nachricht eingesetzt worden, ändert sich das Modell des Adapters und des Hosts. Bei dem empfangenden Adapter entfällt die Warteschlange, da ein weiteres Paket vom Adapter nur akzeptiert wird, wenn er noch einen Pufferbereich frei hat (siehe Kapitel "Protokoll" auf Seite 13 insb.

Set-Flag-9). Die beiden Puffer werden vom Host über seinen Kanal mit der entsprechenden Kanalübertragungsrate geleert. Im Host werden die Pakete in ein von Netex reservierten Hauptspeicherbereich geschrieben, um dann auf eine Magnetplatte (CRAY DD29, IBM 3380) transferiert zu werden. Bei diesem Modell wird die Datenübertragung bis zum Schreiben der Dateien in den Hauptspeicher des empfangenden Host betrachtet. Dieser Schreibvorgang der Daten in den Arbeitsspeicherbereich läuft mit einer Geschwindigkeit ab, die größer oder gleich der Kanalgeschwindigkeit ist. Unter einer solchen Voraussetzung entfällt eine Warteschlange im empfangenden Host, und die Datenübertragungsrate wird nur durch die Kanalgeschwindigkeit des Hosts bestimmt.



In dem verwendeten Modell werden folgende vereinfachende Annahmen gemacht:

1. Es kann nur wahlweise eine Message-With-Data- oder eine Message-Only-Sequenz betrachtet werden (siehe Kapitel "Frames und Framesequenzen" auf Seite 18). Da die Untersuchung des File-Transfers im HYPERchannel/50 in dieser Arbeit im Vordergrund steht, wird das Augenmerk auf die Message-With-Data-Sequenz gelegt, welche für den Datentransport verantwortlich ist.
2. Die Message-With-Data-Sequenz wurde in dem Modell modifiziert dargestellt, um einen überschaubareren Ablauf der Aktivitäten auf dem Trunk zu erhalten (Abbildung 16).



In dem Modell wurden drei unabhängige Phasen zu einer Phase zusammengefaßt. Bei den drei Phasen handelt es sich um die Reservierungsphase (R), bei welcher der Sender versucht den Empfänger zu belegen, um die Übertragung des Flag A, welches dem Empfänger anzeigt, daß Daten fol-

gen, sowie um das Flag 9, welches die Bereitschaft des Empfängers zur Datenaufnahme signalisiert. Der Aufbau eines Paketes, welches im HYPERchannel/50-Protokoll aus vier Teilstücken besteht, wobei zwei Verzögerungsphasen und zwei Übertragungsphasen durchlaufen werden, ist in dem Modell auf eine Verzögerungsphase und eine Übertragungsphase reduziert worden. Die Gesamtzeit für ein Paket ist mit 487 Mikrosekunden beibehalten worden und das Verhältnis von Daten zu Verzögerung ist mit 87% zu 13% auch unverändert geblieben. Die Schlußphase, bei der die virtuelle Verbindung abgebaut wird, ist in dem Modell mit 32 Mikrosekunden dargestellt worden, jedoch findet hier keine Übertragung mehr statt.

Die modifizierte Abbildung einer Sequenz enthält keine wesentliche Einschränkung, da die Gesamtzeit für die Sequenz nicht geändert wird, sondern eine Zusammenfassung der vielen einzelnen Übertragungsphasen stattfindet. Das Verhältnis der Zeit bei welcher der Trunk belegt ist, zu der Zeit bei welcher der Trunk freigegeben ist, wird durch die Modifikation nicht berührt.

3. Das Modell berücksichtigt kein 'Error-Recovery' während der Datenübertragung.
4. Ereignisse, die einen Retry-Counter-Overflow erfahren, geben keine Meldung an eine höhere Protokollebene, sondern werden nach einem festen Zeitintervall nochmals ausgeführt. Falls der Retry-Counter eines Adapters abläuft und er nicht erfolgreich war bei seinem Bemühen den gewünschten Empfänger zu reservieren, wird der Adapter freigegeben und das Ereignis nach einer vorher festzulegenden Zeit erneut ausgeführt. Dieses Verhalten weicht von dem im HYPERchannel/50-Protokoll vorgesehen Ablauf ab, da hier eine Meldung an eine höhere Ebene (Layer 6) gegeben wird. Der Fall eines Reservierungsabbruchs tritt sehr selten ein.
5. Es werden nur Konfigurationen mit einem Trunk betrachtet. Der Trunk-Selektions-Algorithmus wird in dem Modell nicht berücksichtigt.

Zur Lösung von Modellen gibt es grundsätzlich zwei Methoden:

1. Analytische Verfahren
2. Simulationsmodelle

Bei analytischen Verfahren versucht man mit den Mitteln der Wahrscheinlichkeitstheorie das vorgegebene Modell zu lösen. Insbesondere für den HYPERchannel/50 wurde schon 1981 eine eingehende Untersuchung des Protokolls und eine Leistungsbewertung mit Hilfe eines 'Offenen-Netzwerk-Modells' durchgeführt /Span 81/.

Analytische Verfahren verlangen jedoch sehr strenge Voraussetzungen bzgl. der Verteilungen (Ankunftsrate, Bedienrate) und der Zuteilungsdisziplinen in dem Modell. Diese Parameter sind in vielen Fällen entweder unbekannt oder man muß aufgrund der Komplexität der Berechnungen vereinfachende Annahmen machen. Weiterhin können durch analytische Verfahren parallele Prozesse wie die Wechselpuffer in den Adaptern nur schwer erfaßt werden. Aus diesen Gründen ist in dieser Arbeit zur Abbildung des HYPERchannel/50-Protokolls ein Simulationsmodell gewählt worden.

Simulationsprogramme können während verschiedener zeitlicher Phasen des realen Systems eingesetzt werden, d.h. sie haben ihre Berechtigung sowohl vor dem Einsatz des realen Systems, als auch während und nach Inbetriebnahme des abgebildeten Systems.

Eine wichtige Aufgabe der Simulation ist die Möglichkeit Vorhersagen über das Systemverhalten zu liefern, bevor dieses System in der Realität eingesetzt oder getestet wurde. Auf diese Weise erhält man Vorhersagen über die Leistungsfähigkeit und das Verhalten der simulierten Realität. Besonders bei aufwendigen und teuren Projekten kann man mit Hilfe der Simulation Erkenntnisse und Einblicke in das simulierte System erhalten, ohne daß das reale System benutzt werden müßte. Die Simulation führt in solchen Fällen zu einer besserer Einsicht und einem tieferen Verständnis des untersuchten Projekts.

Sobald ein Simulationsprogramm eines realen Systems besteht, können Messungen und Untersuchungen des Verhaltens des abgebildeten Modells vorgenommen werden. Dabei können verschiedene Meßpunkte definiert werden und statistische Aussagen gewonnen werden. Auf diese Weise lassen sich Schwachstellen und Engpässe herausfinden und durch Variation der Systemparameter günstigere Konfigurationen ermitteln. Diese Messungen und Variationen können im Simulationsprogramm durchgeführt werden, ohne daß das reale System gestört wird oder verändert werden muß.

Die Güte eines Simulationsprogramms ergibt sich zumeist durch die Exaktheit der abgebildeten Realität. Hier können folgende Faktoren hineinspielen:

- Kenntnis und Unterlagen des zu modellierenden Systems,
- Wahl des Programmsystems bzw. der Programmiersprache,
- verfügbare Rechnerleistung etc.

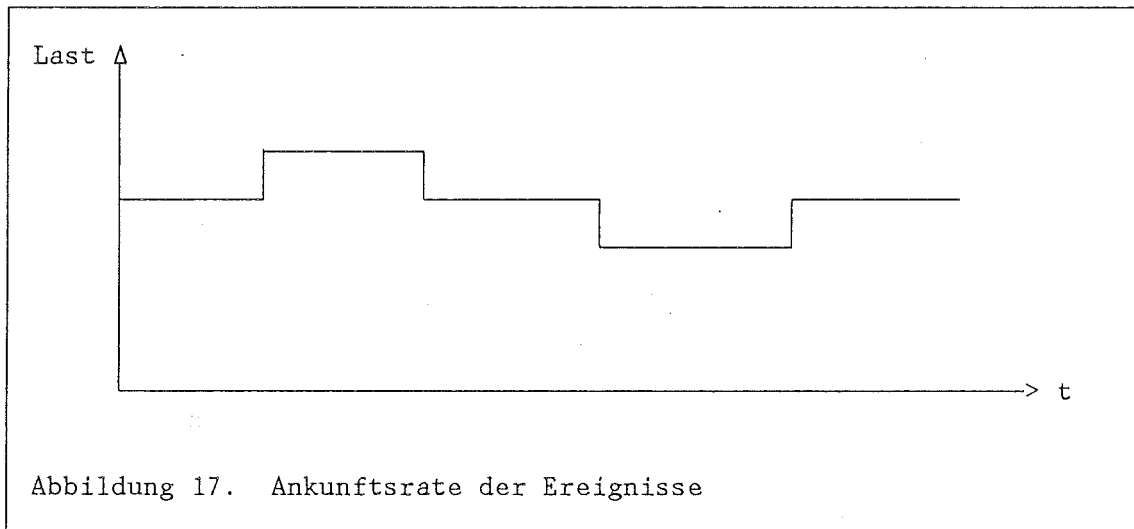
Die Grenzen des Simulationsprogramms werden somit neben dem Verständnis des realen Systems durch den vorhandenen Rechner bestimmt. Je aufwendiger das Modell ist, desto höher wird die benötigte Rechnerleistung.

Man kann zwischen zwei Formen von Simulationsmodellen unterscheiden:

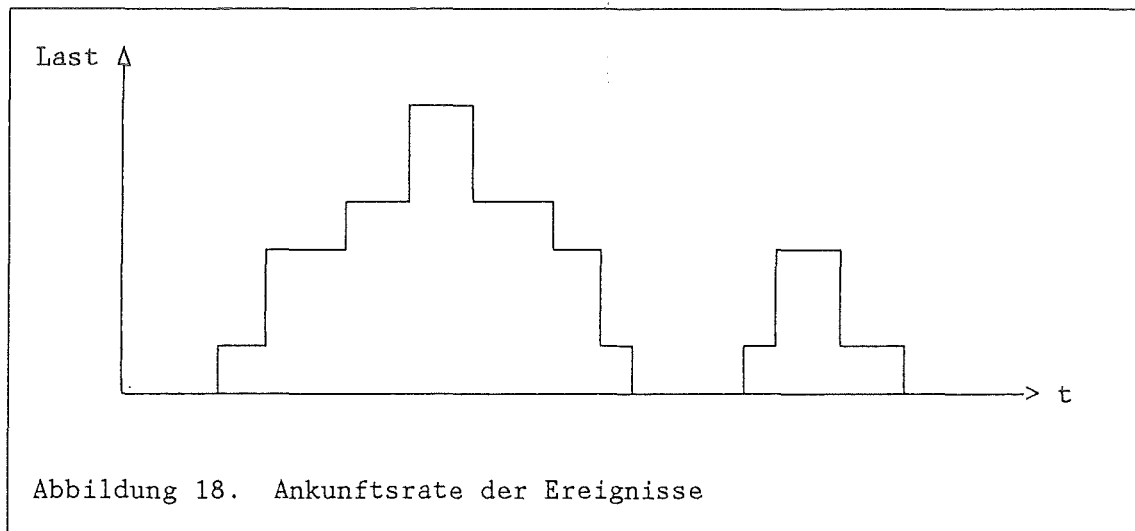
1. Time-Driven (Zeit-getriebene Modelle)
2. Event-Driven (Ereignis-getriebene Modelle)

Bei den Zeit-getriebenen Modellen wird ein festes Zeitintervall gewählt, um das die Simulation fortschreitet. Bei jeder Erhöhung der Simulationszeit, wird das Modell in einen entsprechenden Zustand versetzt. Dieses Modell ist gut geeignet, falls eine gleichmäßige Verteilung der auftretenden Ereignisse der Simulation zugrunde liegt (Abbildung 17 auf Seite 41), wie dies beispielsweise für die Simulation von Speicherreferenzierungen gilt, bei der die Taktrate des Rechners naheliegenderweise ein Zeitintervall impliziert. Bei stark schwankenden zeit-

lichen Abständen zwischen den Ereignissen, d.h. die Zwischenankunftszeit variiert erheblich, hat dieses Modell jedoch den Nachteil, daß die wichtigen Phasen nur ungenau dokumentiert werden und die unwesentlichen Zeitabschnitte, bei denen nichts passiert, zu ausführlich festgehalten werden.



Ereignis-getriebene Verfahren simulieren die Realität, indem sie für jede Änderungen des Systemzustandes ein Ereignis vorsehen und die Simulationszeit immer um den zeitlichen Abstand zu dem vorhergehenden Ereignis erhöhen. Diese Modelle sind somit stark lastabhängig (Abbildung 18 auf Seite 42) und können sowohl sehr aufwendig werden - falls viele Ereignisse in kurzer Zeit eintreten - als auch einfach werden - falls kaum oder gar keine Ereignisse auftreten.



Für die Simulation des HYPERchannel/50 wurde ein Ereignis-getriebenes Modell gewählt, da durch die Simulation Zeiten von 1 Mikrosekunde, für die Ausbreitung des elektrischen Signals im Trunk, bis zu etwa 10 Sekunden, für die Betrachtung eines File-Transfers abgedeckt werden müssen. Es ergeben sich somit Zeitdifferenzen bis zu einem Faktor von 10^7 , die durch ein Ereignis-getriebenes Modell effizienter berücksichtigt werden können.

Existierende Simulationsmodelle

Der HYPERchannel/50 war Gegenstand mehrerer unterschiedlicher Simualtionsmodelle und es bestehen bisher die folgenden Untersuchungen:

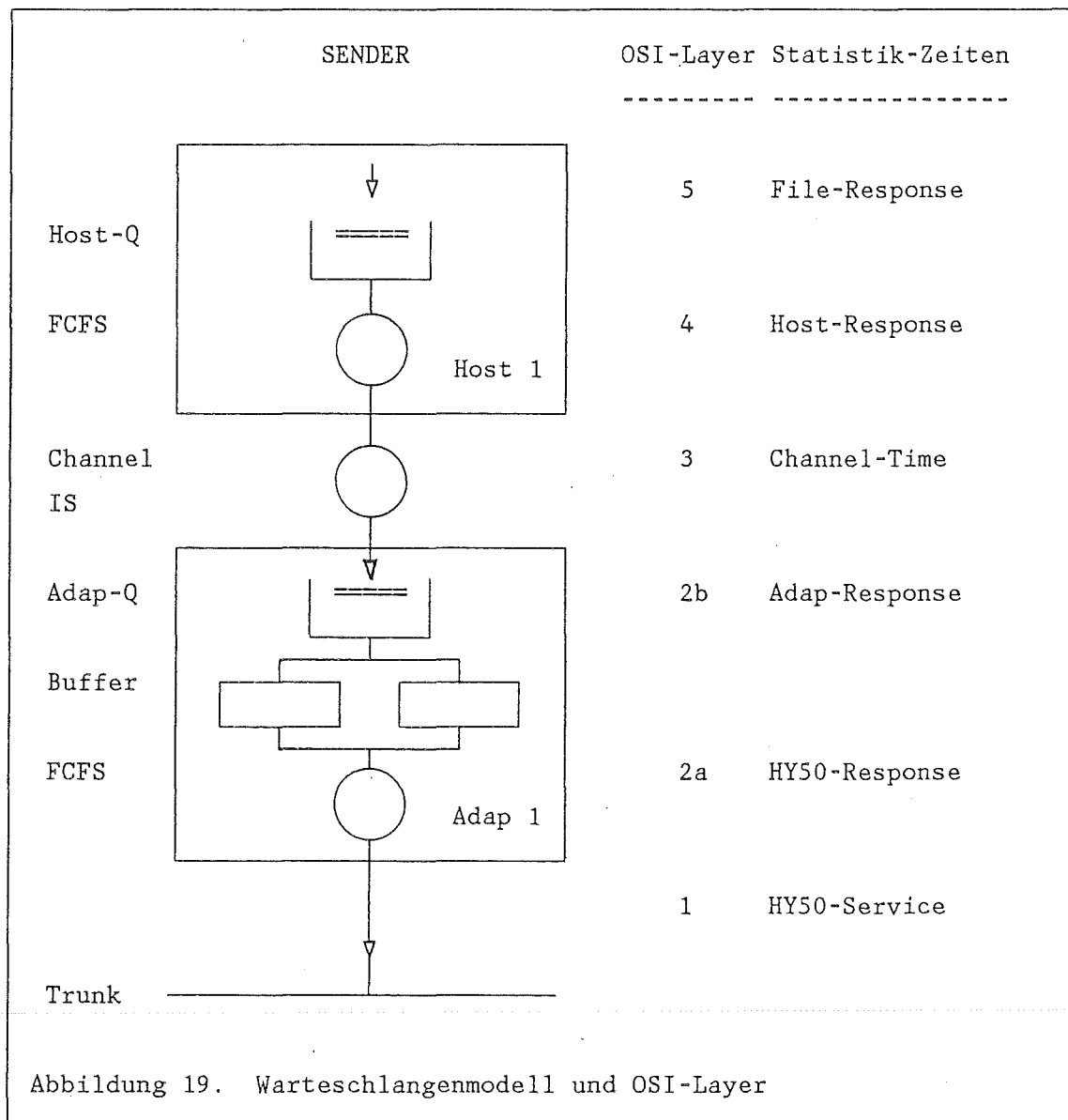
1. Das erste Simulationsmodell wurde in /Donn 79/ und in /Fran 80/ beschrieben und betrachtet ein Netz mit einem Trunk und drei Adaptern, wobei der mittlere Adapter die höchste Priorität erhält. (In der Praxis würde man, um die Verzögerungen zu minimieren, die Prioritäten in der Reihenfolge vergeben, in der die physische Anordnung der Adapter an dem Trunk vorgenommen wurde.) Auf Grund der geringen Adapteranzahl konnten jeweils nur eine Übertragung stattfinden, was dazu führte, daß keine Auswirkungen des Wait-Flip-Flops und des Prioritätenschemas festgestellt werden konnten. Auch ließen sich keine Messungen bzgl. der effektiven Datenrate, des maximalen Durchsatzes und zu dem Antwortzeitverhalten unter Last durchführen.

2. Das zweite Modell wurde in /Katk 81/ vorgestellt, jedoch wurden keine Ergebnisse angegeben. Die Beschreibung des Modells läßt den Schluß zu, daß starke Vereinfachungen gegenüber dem HYPERchannel/50-Protokoll vorgenommen wurden.
3. Das entscheidende Simulationsmodell wurde von W.R. Franta und J.R. Heath 1982 vorgestellt /Fran 82/ und 1984 modifiziert /Fran 84/. In diesem Modell wurde das HYPERchannel/50-Protokoll sehr exakt und umfassend abgebildet. Dabei ist das Trunk-Zugriffsverfahren, das Link-Level-Protokoll, der virtuelle Verbindungsaufbau und das Trunk-Auswahlverfahren beschrieben und untersucht worden. Insbesondere wurde auf die Auswirkungen des Wait-Flip-Flops, der unterschiedlichen Puffergröße (2 oder 4 Kbyte) und den Durchsatz bei den beiden unterschiedlichen Sequenztypen (Message-Only und Message-With-Data) genau eingegangen. Interessanterweise wurde auch die Besonderheit des HYPERchannel/50 bis zu vier Leitungen an jeden Adapter anschließen zu können in den Untersuchungen berücksichtigt /Fran 85/.

Die Untersuchungen beschränkten sich jedoch im wesentlichen auf ein Netzwerk mit sechs angeschlossenen Adapter, wovon 3 Adapter als Sender und drei Adapter als Empfänger konfiguriert waren. Dabei schickte jeder Sender nur an seinen fest zugeteilten Empfänger, der in minimalem Abstand von 13 m direkt neben ihm positioniert ist. Diese Konfiguration ist für die Untersuchung des Trunk-Zugriff-Verfahrens ausgelegt, wobei die Anzahl der erfolglosen Sendungsversuche und die Schaltung des Wait-Flip-Flops sehr deutlich gemessen werden können. Allerdings besteht bei diesem simulierten Netz keine Möglichkeit das Verhalten des Reservierungsschemas bei gleichen Sendern bzw. Empfängern zu untersuchen. Die virtuelle Adapterverbindung und die Auflösung der Request-Reserve-Loops ist jedoch ein wichtiger Bestandteil des Protokolls. Weiterhin wurde keine Verzögerung durch die begrenzte Kanalgeschwindigkeit zwischen dem Host und seinem Adapter berücksichtigt.

6.0 PROGRAMM

Bei dem in dieser Arbeit verwendeten Simulationsprogramm handelt es sich um ein Ereignis-getriebenes Programm, welches in Anlehnung an ein von Chandy und Sauer /Chan 84/ vorgestelltes Verfahren zur Implementierung einer Simulation entwickelt wurde.



Das Programm wurde in drei Stufen aufgebaut. Der Grund dieser Unterteilung war die Intention parallel zu dem ISO-OSI-Referenzmodell im Programm die

einzelnen Schichten nachzubilden. Insbesondere die Physikalische- und die Verbindungsebene, die bei dem HYPERchannel/50 in den Adaptern implementiert sind, sollte streng von den höheren Ebenen, welche durch die Software (Netex) in dem Host abgewickelt wird, getrennt werden. Dies führte zu einem hierarchischen Programmaufbau und einem stufenweisen 'Bottom-Up' Vorgehen.

6.1 PROGRAMMSTRUKTUR

Zunächst wurden die Ebenen 1 und 2a ('Medium Access Control') untersucht, wobei die Ausbreitung des Signals im Medium und der 'Window'-Mechanismus des Prioritätenverfahrens betrachtet wurde. In dem darauffolgenden Schritt ist der Aufbau eines Datenpaketes und das Reservierungsverfahren analysiert worden (Layer 2b und 3). Zum Schluß wurde ein File-Transfer, der den Host und den Kanal zwischen Adapter und Host mit einschließt, untersucht. Diese Stufe umfasst die beiden Ebene 4 und 5. In der Statistik des Programms werden die Zeiten für die einzelnen OSI-Layer getrennt festgehalten und in Antwort- bzw. Servicezeit unterteilt (siehe Kapitel "Ausgabe" auf Seite 50). Die Kommunikation zwischen diesen unabhängigen Ebenen erfolgt über eine vordefinierte und feststehende Schnittstelle.

Das Programm ist in der Sprache Pascal geschrieben worden und bedient sich der in Pascal zur Verfügung stehenden Strukturelemente, insbesondere der Records zur Abbildung eines Ereignisses und der Pointer zum Aufbau der Warteschlangen. Die Gründe für die Verwendung von Pascal als Programmiersprache - und nicht Simula, GPSS oder RESQ etc. - liegt darin begründet, daß Pascal die vom Autor an besten beherrschte Sprache ist, und durch das Buch von K.M. Chandy und C.H. Sauer /Chan 84/ existierte eine gute Grundlage für Simulationsprogramme.

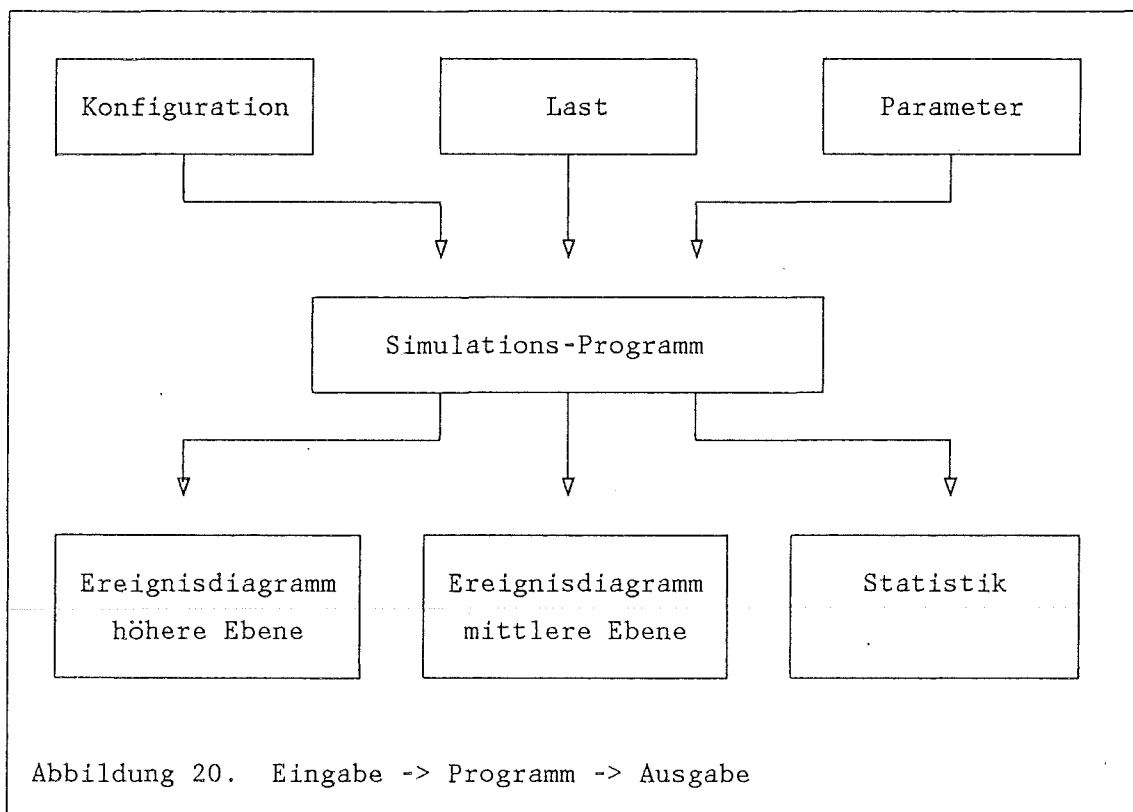
In dem Programm werden alle Ereignisse unabhängig davon, ob sie als Startwerte eingegeben wurden, oder ob sie zur Laufzeit der Simulation erzeugt wurden, in die Ereigniswarteschlange eingereiht. Diese Warteschlange ist aufsteigend bzgl. der Zeitpunkte des Eintritts der Ereignisse geordnet und wird mit Hilfe einer Prozedur 'Removevent' abgearbeitet. Den Kern des

Simulationsprogrammes bildet somit die Warteschlange für die Ereignisse und die Prozeduren 'Insertevent', 'Removevent' und 'Handlevent'. Alle Ereignisse werden durch festgelegte Ereignistypen in einem Case-Fall getrennt und unterschiedlich abgearbeitet. Es stehen folgende Ereignistypen zur Verfügung:

- Host-Event
- Mess-Data-Event
- Physical-Event

Jedes 'Event' stellt ein Ereignis der entsprechenden Programmebene dar, und erlaubt auf diese Weise die getrennte Abbildung der OSI-Layer.

Das Simulationsprogramm erhält seine Eingabedaten aus drei unterschiedlichen Dateien und erzeugt drei unterschiedliche Formen der Ausgabe (Abbildung 20).



6.2 EINGABE

Für das Simulationsprogramm sind drei Formen der Dateneingabe vorgesehen.

1. Datenliste
2. Parameterliste
3. Konfigurationsliste

Diese Trennung wurde vorgenommen, da die Listen unterschiedliche Arten von Eingabedaten enthalten (Abbildung 21 auf Seite 49). Während die Werte der Datenliste die Ereignisse bilden, mit denen der Sendeadapter, der Empfangsadapter, die Startzeit und die Dateigröße festgelegt wird, und damit die Last erzeugen, enthält die Parameterliste Daten für den Aufbau der Message-With-Data-Sequence, der Blockgröße, der Schaltung des Wait-Flip-Flop's (On/Off), die Retry-Zeit und das Retry-Limit, sowie eine Verzögerung für die Ereignisse, deren Retry-Counter abgelaufen ist (siehe Kapitel "Modell" auf Seite 35). In der Konfigurationsliste wird die Ausbreitungsgeschwindigkeit des elektrischen Signals im Trunk, die Anordnung der Hosts an dem HYPERchannel/50 und die Kanalgeschwindigkeiten der angeschlossenen Hosts festgelegt.

a) DATENLISTE :

From-Adap	To-Adap	Start-Zeit	Filesize (KB)	Sonst
-----	-----	-----	-----	-----
1	2	100	10	1 1

b) PARAMETERLISTE :

Prepare-Self	Reserve-Partner	Delay	Data-Block	Delay
-----	-----	-----	-----	-----
64	14	193	424	63

Blocksize	Wait-F-F	Retry-Time	Retry-Limit	Softdelay
-----	-----	-----	-----	-----
2	0	1000	3	5000

c) KONFIGURATIONSLISTE :

Prop-Delay	Distanz-1	Distanz-2	Distanz-3	Distanz-4
-----	-----	-----	-----	-----
0.004	111	111	111	

Kanal 1	Kanal 2	Kanal 3	Kanal 4	Higrafic Time (dt)
-----	-----	-----	-----	-----
3000	3000	1500	1500	500

Abbildung 21. Eingabedaten für das Simulationsprogramm

Neben diesen genannten Parametern kann die Form der Ausgabe (Ereignisdiagramm der höheren Ebene, Ereignisdiagramm der mittleren Ebene, Statistik) zum Zeitpunkt des Starts der Simulation gewählt werden (siehe Kapitel "Ausgabe" auf Seite 50). Die Anzahl der angeschlossenen Adapter kann nicht zur Laufzeit des Programmes bestimmt werden, da dieser Wert als Konstante

in dem Konstantendeklarationsteil des Pascal-Programms zur Compile-Zeit eingegeben werden muß. Zur Zeit ist in dem Simulationsprogramm eine obere Grenze von sechs Adaptern vorgesehen, die durch die Größe der Ausgabediagramme bestimmt wird.

6.3 AUSGABE

Das Simulationsprogramm liefert drei unterschiedliche Formen der Ausgabe. Diese Unterteilung wurde in Hinsicht auf die angestrebte Äquivalenz zu dem ISO-OSI Modell gewählt, wobei das Ereignisdiagramm der mittleren Ebene die Adapteraktivität der Ebene 2b (CSMA/CA) widerspiegelt und das Ereignisdiagramm der höheren Ebene einen File-Transfer aus der Sicht der Ebenen 4 bzw. 5 (Kommunikationsverbindung, Auf- und Abbau der Sitzungen) darstellt.

Statistik

Die Statistik bildet die Grundlage für die verwendeten Diagramme. (Abbildung 22 auf Seite 51). Sie ist in zwei Teile untergliedert, um für beide betrachteten Ebenen unabhängige Werte zu liefern.

1. Host - Statistik
2. Adapter - Statistik

SIMULATION - STATISTICS for HOST:

Number of Events : 1 Simulated Time : 6.50 ms

Device: Completed - Jobs : Throughput : Q-Length: Q-Time:

to HC	send	rece	total	Sending	Receive	Total	av.	av.(ms)
Trunk	1	1	2	0.15	0.15	0.31		
Host1	0	0	0	0.00	0.00	0.00	0.00	0.00
Host2	1	0	1	0.15	0.00	0.15	0.58	3.75
Host3	0	1	1	0.00	0.15	0.15	0.00	0.00
Host4	0	0	0	0.00	0.00	0.00	0.00	0.00

Device: Busy-T: Busy-T: Servic: Expans: % Over: Respon: Servic:

to HC	(ms)	(%)	total	factor	q,u,t,k	av.(ms)	av.(ms)
Trunk	1.3	20					
Host1	0.0	0	0.0	0.0	0	0.00	0.00
Host2	3.8	58	3.6	1.1	5	3.75	3.55
Host3	5.2	80	5.2	1.0	0	5.20	5.20
Host4	0.0	0	0.0	0.0	0	0.00	0.00

SIMULATION - STATISTICS for Adapter:

Device: Completed - Jobs : Throughput : Q-Length: Q-Time:

to HC	send	rece	total	Sending	Receive	Total	av.	av.(ms)
Trunk	3	3	6	0.46	0.46	0.92		
Adap1	0	0	0	0.00	0.00	0.00	0.00	0.00
Adap2	3	0	3	0.46	0.00	0.46	0.27	0.58
Adap3	0	3	3	0.00	0.46	0.46	0.00	0.00
Adap4	0	0	0	0.00	0.00	0.00	0.00	0.00

Device: Busy-T: Busy-T: Servic: Expans: % Over: Respon: Servic:

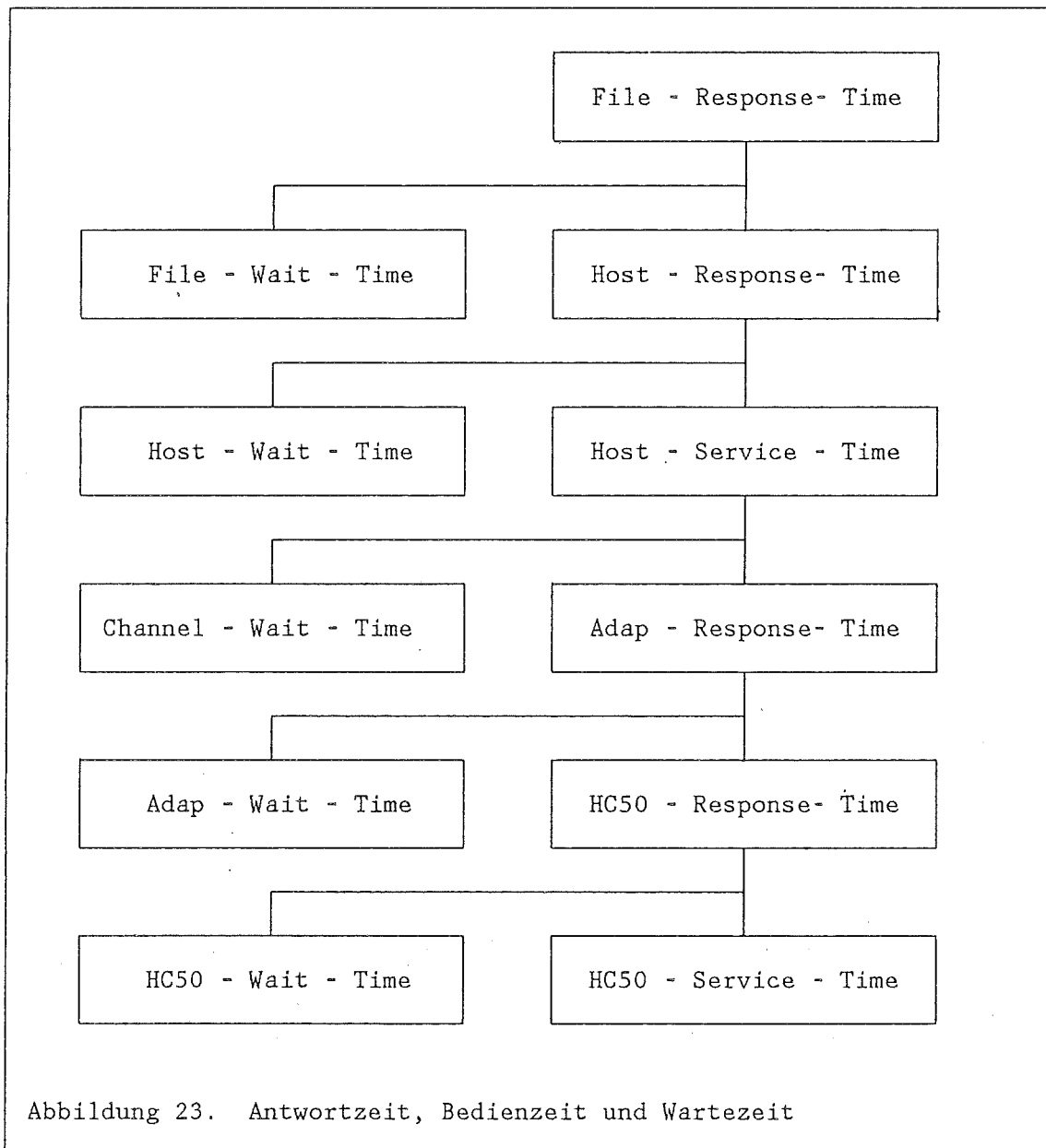
to HC	(ms)	(%)	total	factor	q,u,t,k	av.(ms)	av.(ms)
Trunk	1.3	20					
Adap1	0.0	0	0.0	0.0	0	0.00	0.00
Adap2	1.7	27	1.7	1.0	0	0.58	0.58
Adap3	1.5	23	1.5	1.0	0	0.50	0.50
Adap4	0.0	0	0.0	0.0	0	0.00	0.00

Abbildung 22. Statistik eines Simulationslaufes

In beiden Teilen der Statistik werden Summen gebildet und Mittelwerte errechnet:

- Anzahl der vollendeten Jobs (Number of Events);
- Durchsatz der vollendeten Jobs (Throughput);
- mittlere Warteschlangenlänge (average Queue-Length);
- mittlere Zeit des Jobs in der Warteschlange (average Queue-Time);
- Busy-Zeit des Host bzw. Adapters in Millisekunden und Prozent;
- Bedienzeit in Millisekunden (Total Service-Time);
- Expansionsfaktor (Busy-Time/Utility-Time);
- Overhead $((\text{av. Responsetime} - \text{av. Servicetime}) / \text{av. Responsetime} * 100)$;
- Mittlere Antwortzeit für Datei bzw. Block (av. Response-Time);
- Mittlere Bedienzeit für Datei bzw. Block (av. Service-Time);
- Auslastung des Trunks (in ms und prozentual);

Bei der Größe 'Q-Time' handelt es sich beispielsweise um die mittlere Wartezeit, die ein Ereignis der entsprechenden Ebene (Adapter oder Host) in der Warteschlange verbringt. In der Spalte 'Service-Time' wird für den Adapter die mittlere Bedienzeit in ms angegeben, die für ein Datenpaket anfällt. Das Beispiel zeigt deutlich, daß für den Adapter keine Wartezeit anfällt, da die Antwortzeit mit der Bedienzeit übereinstimmt (Adapter 2: 0.50 ms; Adapter 3: 0.58 ms).



Mit Hilfe der Statistik läßt sich eine Aufschlüsselung der Antwortzeit in die Bedienzeit und die Wartezeit an den verschiedenen Ebenen vornehmen. Als Ebenenmodell sollte das in dem Kapitel "Programm" auf Seite 45 angedeutete, und in Abbildung 23 dargestellte Schema dienen. Für das verwendete Beispiel eines File-Transfers von dem Hosts 2 an den Host 3, ergibt sich die in Abbildung 24 auf Seite 54 ausgedruckte prozentuale Verteilung der insgesamt benötigten Zeit auf die verschiedenen Ebenen.

PROZENT - Statistik:	Host 1	Host 2	Host 3	Host 4
-----	-----	-----	-----	-----
File Resp:		100	100	
File-Wait Host-Resp:		0 100	100	
Host-Wait Host-Serv:		5 95	100	
Chan-Wait Adap-Resp:		48 47	71 29	
Adap-Wait HYPC-Resp:		0 47	0 29	
HYPC-Wait HYPC-Serv:		0 47	0 29	
Paket-OvH Data-Time:		21 26	11 18	
Abbildung 24. Prozentuale Aufspaltung in Bedien- und Wartezeit				

Man kann erkennen, daß sich bei dem sendenden Host 2 die Antwortzeit für das File zu 95% aus der Bedienzeit des Hosts und zu 5% aus Wartezeit ergibt. Diese Wartezeit entsteht durch die, gegenüber dem Sender langsamere Kanalgeschwindigkeit des empfangenden Hosts, und führt zu einer Verringerung der Übertragungsrate des sendenden Hosts (siehe auch Abbildung 25 auf Seite 55). Die verbleibenden 95% der Zeit lassen sich in die für die Kanalübertragung notwendige Zeit (48%) und die Antwortzeit des Adapters (47%) aufteilen. Der HYPERchannel/50 erzeugt in diesem Beispiel keine Verzögerung. In der letzten Zeile wird eine Aufspaltung der Bedienzeit des Trunks in die für die reinen Daten notwendige Zeit (21%) und in die restliche Zeit (Overhead des Datenpaketes: 26%) vorgenommen.

Die in der Konfigurationsliste festgelegten Werte bestimmen den Aufbau des Netzwerkes, d.h. die Abstände der angeschlossenen Adapter voneinander, sowie die Verteilung der Adapter an dem HYPERchannel/50. Durch Änderung der Propagation-Time von 4 Nanosekunden auf 1.33 Nanosekunden besteht die Möglichkeit die Entfernungsangaben in feet einzugeben. Aus diesen Input-Daten wird die Länge des Trunks errechnet und damit die Ausbreitungsgeschwindigkeit von einem Ende des Netzes bis zum anderen Ende. Danach werden die Zeiten für das End-Delay, die N-Delay's und das Fixed-Delay ermittelt. Diese Zeiten werden einmal original, wie sie errechnet wurden, und einmal gerundet, wie sie im Programm verwendet werden, ausgegeben (siehe dazu auch Abbildung 43 auf Seite 86). Zum Schluß wird ein Array

ausgedruckt, welches die Propagation-Time vom i-ten bis zum j-ten Adapter enthält ($i, j=1 \dots n$).

Ereignisdiagramme für den Host

Die Ereignisdiagramme für die höhere Ebene beschreiben einen File-Transfer aus der Sicht des Sende- und des Empfangshost. Für sie ist nur von Bedeutung, ob der Adapter einen Block aufnehmen kann, oder ob der Adapter einen Block an den Host zu übertragen hat. Das gesamte Reservierungsverfahren sowie der Trunkzugriff ist für den Host unerheblich. Zum besseren Verständnis des Ablaufs einer Übertragung wird zusätzlich die Belegung der Adapterpuffer und der Trunkzustand ausgegeben.

Event-data= From: 2 To: 3 Host-File 6 Time: 100									
TIME :	H1	: A1 :	H2	: A2 :	HC	: A3 :	H3	: A4 :	H4

Speed :	2.0		2.0		Mbyte/s		1.3		1.3
Delay :	1.0	12	1.0	12	ms/Block	12	1.5	12	1.5

0 :
500 :	.	..	2 S3	c.
1000 :	.	..	2 S3	c.
1500 :	.	..	1 S3	tc	B	t.	0 R2	..	.
2000 :	.	..	1 S3	.c	.	c.	0 R2	..	.
2500 :	.	..	0 S3	ct	B	ct	0 R2	..	.
3000 :	.	..	0 S3	c.	.	cw	0 R2	..	.
3500 :	.	..	0 A3	t.	B	tc	1 R2	..	.
4000 :	wc	1 R2	..	.
4500 :	wc	1 R2	..	.
5000 :	c.	2 R2	..	.
5500 :	c.	2 R2	..	.
6000 :	c.	2 R2	..	.
6500 :

Abbildung 25. Ereignisdiagramm für den Host

Die erste Zeile des Diagramms enthält eine Übersicht aller Ereignisse, die zu Beginn der Simulation als Startdaten eingegeben wurden. Diese Daten sind nach den Zeitpunkten des Eintritts der Ereignisse sortiert und bilden den Startzustand der Ereigniswarteschlange. Die nachfolgende Tabelle zeigt für jeden Host mit seinem Adapter je zwei Spalten (Hi,Ai), in welchen der Zustand dieser Komponente zu bestimmten Zeitpunkten markiert ist. Zusätzlich existiert noch eine Spalte (HC) für den Zustand des HYPERchannel/50. Der HYPERchannel/50 kann zwei Zustände annehmen:

1. 'B' : mit der Übertragung einer Nachricht beschäftigt;
2. '.' : frei;

In der Spalte, die den Zustand der Hosts (Hi) abbilden soll und mit H überschrieben ist, findet man einen '.', falls der Host im Moment keine Verbindung mit seinem Adapter aufgebaut hat, d.h. es werden weder Daten an den Adapter gesendet noch Daten empfangen. Wenn der Host nicht frei ist, erscheint zuerst die Nummer des aktuell übertragenen Blocks, der Buchstabe 'S' für Send oder 'R' für Receive sowie die Nummer desjenigen Hosts mit dem er eine virtuelle Verbindung aufgebaut hat. Falls der Host als Sender keine Daten mehr an seinen Adapter schickt, sondern nur noch auf die Bestätigung seines Adapters wartet, daß alle Blöcke korrekt angekommen sind, erscheint ein 'A'. Diese Unterscheidung zwischen 'S' und 'A' zeigt das Ende der Kanalaktivität an und erleichtert die Auffindung des Endes einer Sequenz.

Unmittelbar neben der Hostspalte erscheint die Spalte für den zugehörigen Adapter. Jeder Adapter wird durch die getrennte Darstellung seiner beiden Puffer abgebildet. Die Puffer können die folgenden Zustände einnehmen:

- '.' : Idle
- 'c' : Channel-Activity
- 't' : Trunk-Activity
- 'w' : Wait

In der Tabelle kann man erkennen, daß der Host 2 eine Datei mit 4 Kbyte an den Host 3 schicken möchte. Der Adapter 2 geht nun zuerst in den Zustand 'c' über, da er von seinem Host 2 einen Block mit 2 Kbyte geschickt be-

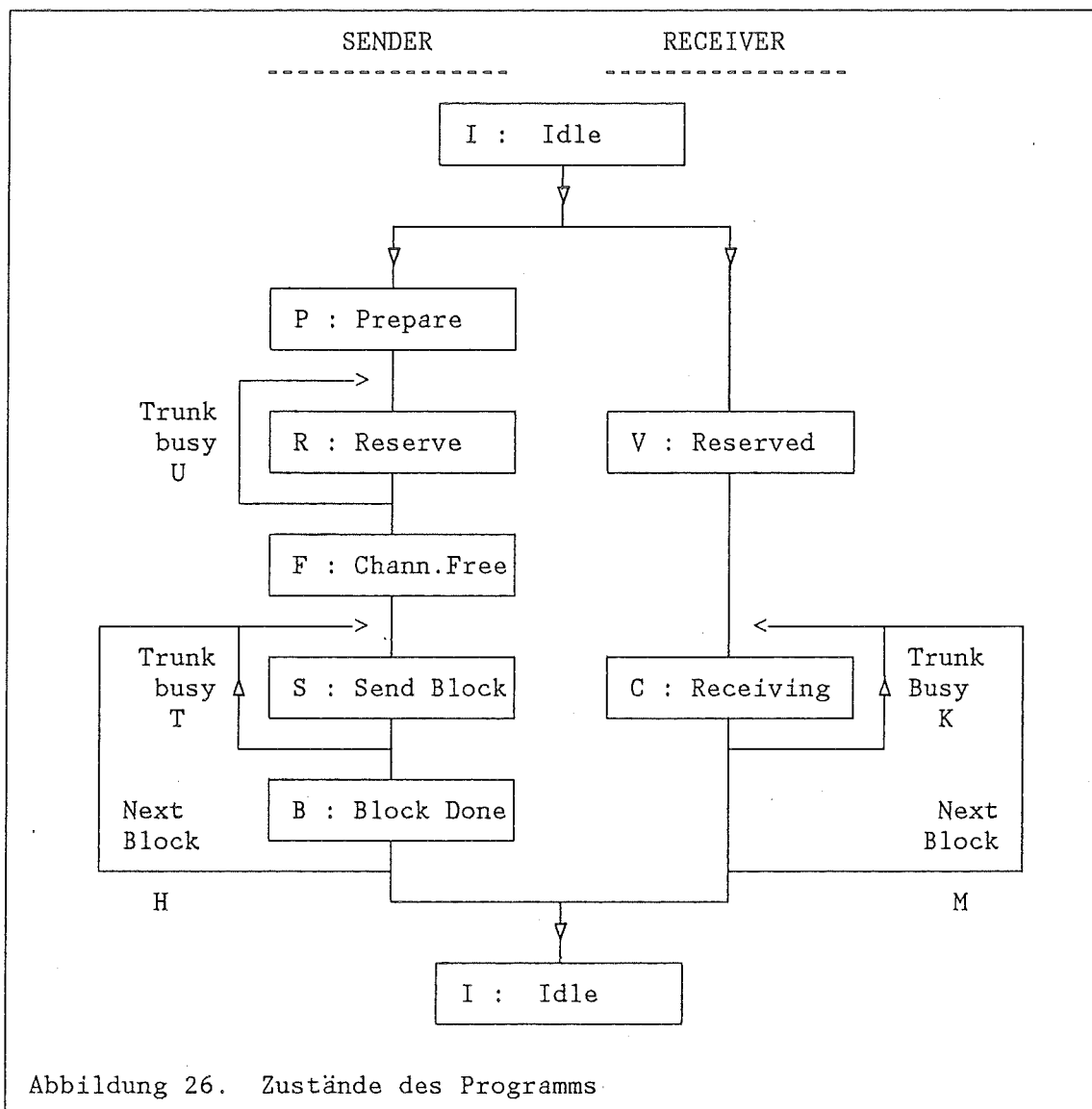
kommt. Sobald dieser Block sich komplett im Puffer des Adapters 3 befindet, kann der Adapter mit der Übertragung des Blocks über den Hyperchannel beginnen. In dieser Phase wird der entsprechende Puffer mit einem 't' gekennzeichnet. Wenn sein gewünschter Empfänger, in diesem Fall der Adapter 3, frei ist kann er für diese Sequenz reserviert werden. Sofort nachdem der Adapter 3 belegt wurde erhält er die ersten Daten vom Sender, d.h. der Adapter 3 wird durch ein 't' dargestellt. Gleichzeitig kann nun der Adapter 2 in seinen zweiten bisher unbenutzten Puffer einen weiteren Block seines Hosts 2 aufnehmen (siehe auch Abbildung 5 auf Seite 13)

An dieser Stelle erkennt man den Vorteil zweier unabhängiger Puffer in jedem Adapter: Es kann parallel ein Block über den Kanal empfangen werden und ein anderer Block über den HYPERchannel/50 übertragen werden. Wäre nur ein Puffer vorhanden, könnte dieser erst wieder freigegeben werden, nachdem der empfangende Adapter den korrekten Eingang dieses Paketes bestätigt hat (Abbildung 24 auf Seite 54).

Zusätzlich enthält das Diagramm für jeden Host seine individuelle Kanalgeschwindigkeit in Mbyte/s. Aus der Kanalgeschwindigkeit resultiert eine Zeitverzögerung, die für ein Paket benötigt wird, um es über den Kanal vom Host zu dem Adapter zu schicken bzw. um es vom Adapter zum Host zu übertragen. Diese Verzögerung ist im Diagramm unterhalb der Kanalgeschwindigkeit aufgeführt und wird in Millisekunden je Block ausgegeben. Das Zeitintervall zu dem der Zustand des gesamten Systems festgehalten wird ist wählbar und wird im weiteren zumeist 500 μ s betragen.

Ereignisdiagramme für den Adapter

Die Diagramme dieser Ebene sollen den zeitlichen Ablauf einer Sequenz während aller Phasen detailliert abbilden. Insbesondere sollen Verzögerungen bei dem Reservierungsverfahren und bei dem Trunkzugriff verdeutlicht werden.



In Anlehnung an eine Message-With-Data-Sequence (Abbildung 16 auf Seite 37) wurden in dem Programm entsprechende Zustände geschaffen, die für den Sender und den Empfänger unterschiedlich sind (Abbildung 26 auf Seite 58). Zu Beginn einer Sequenz befindet sich der Adapter des Senders in dem Zustand idle (I). Nachdem er von seinem Host eine Übertragungswunsch erhalten hat, geht er in den Zustand prepare (P) über und bereitet das Datenpaket vor. Daraufhin wechselt er in den Zustand reserve (R) und versucht den Empfänger zu reservieren. Wenn ihm dies möglich war, ist nun eine virtuelle Verbindung aufgebaut und der Empfänger ist reserved (V). Falls der Trunk belegt war, geht der Sender in den Zustand unable (U) und wartet auf sein N-Delay. Falls der Empfänger schon reserviert war, fährt der Sender mit dem Reservierungsschema fort und befindet sich im Zustand (Q). Im weiteren Verlauf der Sequenz gibt der Sender zunächst den Trunk frei (F) und möchte danach sein Paket übertragen. In dieser Situation findet der Sender entweder den Trunk unbelegt vor, und er geht in den Zustand (S), sowie sein Empfänger in (C), oder der Trunk ist busy, worauf der Sender in den Zustand (T) wechselt und sein Empfänger in (K). Sobald der Block komplett gesendet wurde, geht der Sender in den Zustand block done (D). Falls es sich bei dem zu übertragenden File um mehrere Pakete handelt, wechselt der Sender in der Zwischenzeit in den Zustand host (H) und der Empfänger in (M). Sobald die Datei fertig übermittelt wurde, sind beide wieder idle (I).

Diese weitreichende Auffächerung in verschiedene Zustände ermöglicht die einfache Errechnung von Wartezeit und Servicezeit (Abbildung 27 auf Seite 60). Diese Tabellen dienen somit als Kontrollinstrument für den Ablauf und die Korrektheit des nachgebildeten HYPERchannel/50-Protokolls.

TIME	:	TRUNK	:	Adap.2	:	Adap.3	:	FROM-> TO

757	:	..	:	.P	:	..	:	1 -> 2
	:	..	:	PP	:	..	:	
	:	..	:	PP	:	..	:	
	:	..	:	PP	:	..	:	
821	:	.B	:	PR	:	.V	:	1 -> 2
	:	BB	:	RR	:	VV	:	
837	:	B.	:	RF	:	VV	:	1 -> 2
	:	..	:	FF	:	VV	:	
	:	..	:	FF	:	VV	:	
	:	..	:	FF	:	VV	:	
	:	..	:	FF	:	VV	:	
1030	:	.B	:	FS	:	VC	:	1 -> 2
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
	:	BB	:	SS	:	CC	:	
1458	:	B.	:	SD	:	CM	:	1 -> 2
	:	..	:	DD	:	MM	:	
	:	..	:	DD	:	MM	:	
	:	..	:	DD	:	MM	:	
1521	:	..	:	DH	:	MM	:	1 -> 2

Abbildung 27. Ereignisdiagramm für den Adapter

7.0 FALLSTUDIEN

In diesem Kapitel werden mehrere Fallstudien beschrieben, die einige interessante Aspekte und Sonderfälle des im HYPERchannel/50 verwendeten Protokolls beleuchten sollen. Die Eingabedaten sind so gewählt, daß während der Simulation eine Untersuchung des Trunkzugriffsmechanismus, des Reservierungsschemas und des Verhaltens bei unterschiedlichen Kanalgeschwindigkeiten möglich ist.

7.1 GLEICHE KANALGESCHWINDIGKEIT

Zuerst wird der korrekte Ablauf einer Message-With-Data Sequenz zwischen zwei Hosts mit gleicher Kanalgeschwindigkeit gezeigt (Abbildung 28 auf Seite 62). Mit Kanalgeschwindigkeit wird in diesem Zusammenhang die Übertragungsgeschwindigkeit von Host zu Adapter bezeichnet. Aus dem Diagramm kann man ersehen, daß beide Adapter mit der gleichen Geschwindigkeit gefüllt und geleert werden.

Bei der in diesem Beispiel verwendeten Kanalgeschwindigkeit von 2 Mbyte/s handelt es sich um einen Wert, der im Großrechnerbereich liegt. Für einen Block werden bei dieser Konfiguration 1000 Mikrosekunden benötigt um ihn über den Kanal zu übertragen. Dies bedeutet, daß je Block ca. 2/3 der insgesamt notwendigen Zeit für die Kanalübermittlung anfallen, während weniger als 1/3 der Zeit für die HYPERchannel/50 Übertragung benötigt wird. Gut erkennbar ist auch die parallele Arbeitsweise der Wechselpuffer ('c' und 't').

Event-Data= From: 1 To: 2 Host-File 10 Time: 100																		
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				3.0				3.0
Delay	:	1.0		12		1.0		12		ms/Block				0.7		12		0.7

0	:
500	:	4	S2	c.	
1000	:	4	S2	c.	
1500	:	3	S2	tc		0	R1	t.		B	
2000	:	3	S2	.c		0	R1	c.	
2500	:	2	S2	ct		0	R1	ct		B	
3000	:	2	S2	c.		1	R1	.c	
3500	:	1	S2	tc		1	R1	tc		B	
4000	:	1	S2	.c		2	R1	c.	
4500	:	0	S2	ct		2	R1	ct		B	
5000	:	0	S2	c.		3	R1	.c	
5500	:	0	A2	t.		3	R1	tc		B	
6000	:	.		..		4	R1	c.	
6500	:	.		..		4	R1	c.	
7000	:

Abbildung 28. Sequenz bei gleicher Kanalgeschwindigkeit

7.2 UNTERSCHIEDLICHE KANALGESCHWINDIGKEIT

Als nächstes Experiment ist die Berücksichtigung unterschiedlicher Kanalgeschwindigkeiten vorgesehen. Hierbei können zwei Fälle auftreten:

- Host 3 mit der kleineren Kanalgeschwindigkeit schickt ein File an Host 2, welcher über schnellere Kanäle verfügt (Abbildung 29 auf Seite 63). Dies ist der unkompliziertere Fall, da der Adapter des Hosts 2

die Daten schneller abarbeiten kann als sie ankommen und der Adapter somit zwischen den Paketen den Zustand idle annimmt.

Event-data= From: 3 To: 2 Host-File 8 Time: 100																		
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				1.3				1.3
Delay	:	1.0		12		1.0		12		ms/Block				1.5		12		1.5

0	:
500	:		c.		3 S2		..		.
1000	:		c.		3 S2		..		.
1500	:		c.		3 S2		..		.
2000	:	.		..		0 R3		t.		B		tc		2 S2		..		.
2500	:	.		..		0 R3		c.		.		.c		2 S2		..		.
3000	:	.		..		0 R3		c.		.		.c		2 S2		..		.
3500	:	.		..		1 R3		.t		B		ct		1 S2		..		.
4000	:	.		..		1 R3		.c		.		c.		1 S2		..		.
4500	:	.		..		1 R3		.c		.		c.		1 S2		..		.
5000	:	.		..		2 R3		t.		B		tc		0 S2		..		.
5500	:	.		..		2 R3		c.		.		.c		0 S2		..		.
6000	:	.		..		2 R3		c.		.		.c		0 S2		..		.
6500	:	.		..		3 R3		t.		B		.t		0 A2		..		.
7000	:	.		..		3 R3		c.	
7500	:	.		..		3 R3		c.	
8000	:

Abbildung 29. Unterschiedliche Kanalgeschwindigkeit: 1. Fall

- Der Fall, daß die Kanalgeschwindigkeit des empfangenden Host langsamer ist, als die des sendenden Hosts führt zu einer komplexeren Situation (Abbildung 30 auf Seite 64).

Event-data= From: 2 To: 3 Host-File 8 Time: 100									
TIME :	H1	: A1 :	H2	: A2 :	HC	: A3 :	H3	: A4 :	H4

Speed :	2.0		2.0		Mbyte/s		1.3		1.3
Delay :	1.0	12	1.0	12	ms/Block		1.5	12	1.5

0 :
500 :	.	..	3 S3	c.
1000 :	.	..	3 S3	c.
1500 :	.	..	2 S3	tc	B	t.	0 R2	..	.
2000 :	.	..	2 S3	.c	.	c.	0 R2	..	.
2500 :	.	..	1 S3	ct	B	ct	0 R2	..	.
3000 :	.	..	1 S3	c.	.	cw	0 R2	..	.
3500 :	.	..	0 S3	tc	B	tc	1 R2	..	.
4000 :	.	..	0 S3	.c	.	wc	1 R2	..	.
4500 :	.	..	0 S3	.w	.	wc	1 R2	..	.
5000 :	.	..	0 A3	.t	B	ct	2 R2	..	.
5500 :	cw	2 R2	..	.
6000 :	cw	2 R2	..	.
6500 :c	3 R2	..	.
7000 :c	3 R2	..	.
7500 :c	3 R2	..	.
8000 :

Abbildung 30. Unterschiedliche Kanalgeschwindigkeit: 2. Fall

Sobald die virtuelle Verbindung zwischen den beiden Adaptern der beteiligten Hosts aufgebaut ist, wird der erste Block vom Sender an den Empfänger übertragen. Dieser Block findet, genauso wie der zweite Block, in den Puffern des Empfängeradapters Platz und wird von dort mit Kanalgeschwindigkeit an den Host 3 übertragen. Erst wenn der komplette Block vom Host 3 empfangen und quittiert wurde, kann der Puffer vom Adapter 2 für das nächste Paket freigegeben werden. Dies geschieht durch die Übertragung eines Clear-Flag-9 Frames an den Sender, um ihm mitzuteilen, daß der Empfänger bereit ist einen weiteren Block aufzunehmen. Nun erst kann der im Sender-Adapter zur Verfügung stehende

nächste Block via HYPERchannel/50 abgeschickt werden. Nachdem der Block beim Empfänger korrekt angekommen ist und quittiert wurde, gibt der Sender-Adapter diesen Puffer frei und initiiert die Übertragung eines weiteren Paketes vom Host 2. Es zeigt sich, daß, wie immer bei solchen voneinander abhängigen 'Situationen', die resultierende Übertragungsrate durch die langsamste Komponente in der Kette bestimmt wird. In diesem Fall ist dies der langsame Kanal des empfangenden Hosts 3, der mittels des HYPERchannel/50-Protokolls die Datenrate bestimmt.

7.3 HALF-DUPLEX ÜBERTRAGUNG

Als drittes Experiment ist der Fall eines Reservierungswunsches an den sendenden Adapter einer bestehenden Sequenz durch einen dritten Adapter gewählt worden. Während die virtuelle Verbindung zwischen 1 und 2 besteht, versucht der Adapter 3 den Adapter 1 als seinen Empfänger zu reservieren (Abbildung 31 auf Seite 66). Er erhält jedoch von 1 einen negativen Bescheid, worauf im Adapter 3 das Retry-Verfahren zum Zuge kommt. Sobald der sendende Adapter 1 seine Sequenz beendet hat, gibt er sich selbst frei und kann bei nächster Gelegenheit vom Adapter 3 belegt werden.

```

Event-data= From: 2 To: 3 Host-File 6 Time: 100
Event-data= From: 1 To: 2 Host-File 6 Time: 1100
TIME : H1 : A1 : H2 : A2 : HC : A3 : H3 : A4 : H4
-----
Speed : 2.0          2.0          Mbyte/s          1.3          1.3
Delay : 1.0  12      1.0  12  ms/Block          1.5  12      1.5
-----
0 : .  ..      .  ..      .  ..      .  ..      .
500 : .  ..      2 S3  c.      .  ..      .  ..      .
1000 : .  ..      2 S3  c.      .  ..      .  ..      .
1500 : 2 S2  c.      1 S3  tc      B      t.      0 R2  ..      .
2000 : 2 S2  c.      1 S3  .c      .      c.      0 R2  ..      .
2500 : 2 S2  t.      0 S3  ct      B      ct      0 R2  ..      .
3000 : 2 S2  t.      0 S3  c.      .      cw      0 R2  ..      .
3500 : 2 S2  t.      0 A3  t.      B      tc      1 R2  ..      .
4000 : 2 S2  t.      .  ..      .      wc      1 R2  ..      .
4500 : 1 S2  tc      0 R1  t.      B      wc      1 R2  ..      .
5000 : 1 S2  .c      0 R1  c.      .      c.      2 R2  ..      .
5500 : 0 S2  ct      0 R1  ct      B      c.      2 R2  ..      .
6000 : 0 S2  c.      1 R1  .c      .      c.      2 R2  ..      .
6500 : 0 A2  t.      1 R1  tc      B      ..      .  ..      .
7000 : .  ..      2 R1  c.      .      ..      .  ..      .
7500 : .  ..      2 R1  c.      .      ..      .  ..      .
8000 : .  ..      .  ..      .      ..      .  ..      .

```

Abbildung 31. Half-Duplex Übertragung

An dieser Stelle wird deutlich, daß es sich bei dem verwendeten Verfahren nicht um eine Full-Duplex Übertragung handelt, bei der gleichzeitig Pakete ankommen und gesendet werden. Der Adapter ist nicht in der Lage gleichzeitig einen Sende- und einen Empfangswunsch zu bedienen. Vielmehr muß jede Nachricht von (zu) dem Host warten, falls noch eine Übertragung zu (von) dem Host stattfindet. Erst wenn die laufende Sequenz beendet ist, kann eine Übertragung in die andere Richtung begonnen werden. Aus diesem Grund wird das verwendete Schema als Half-Duplex bezeichnet /Ness 81/.

Falls die Übertragung zwischen 2 und 3 zu lange dauert, läuft der Retry-Counter des Adapters 1 ab, da er nicht erfolgreich war bei seinem Versuch den gewünschten Empfänger zu reservieren.

7.4 WARTESCHLANGENBENUTZUNG DER HÖHEREN EBENE

Die vierte Untersuchung beschäftigt sich mit der Organisation durch die Netex-Software (Abbildung 32 auf Seite 68). Falls eine Sequenz zwischen Adapter 1 und Adapter 3 besteht und eine weitere Verbindung von Host 1 mit einem anderen Adapter aufgebaut werden soll, wird dieser Request von der Netex-Software, die auf dem Host 1 abläuft, entgegengenommen und in eine Warteschlange eingereit. Erst wenn die laufende Übertragung zwischen 1 und 3 beendet ist, wird der wartende Übertragungswunsch aktiviert und ausgeführt.

Event-data= From: 1 To: 3 Host-File 4 Time: 100																		
Event-data= From: 1 To: 2 Host-File 6 Time: 1100																		
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				1.3				1.3
Delay	:	1.0		12		1.0		12		ms/Block				1.5		12		1.5

0	:
500	:	1 S3		c.	
1000	:	1 S3		c.	
1500	:	0 S3		tc		.		..		B		t.		0 R1		..		.
2000	:	0 S3		.c			c.		0 R1		..		.
2500	:	0 A3		.t		.		..		B		ct		0 R1		..		.
3000	:	2 S2		c.			cw		0 R1		..		.
3500	:	2 S2		c.	c		1 R1		..		.
4000	:	1 S2		tc		0 R1		t.		B		.c		1 R1		..		.
4500	:	1 S2		.c		0 R1		c.		.		.c		1 R1		..		.
5000	:	0 S2		ct		0 R1		ct		B	
5500	:	0 S2		c.		1 R1		.c	
6000	:	0 A2		t.		1 R1		tc		B	
6500	:	.		..		2 R1		c.	
7000	:	.		..		2 R1		c.	
7500	:

Abbildung 32. Warteschlangenbenutzung der höheren Ebene

Die im Kapitel "Virtuelle Adapter-Verbindungen" auf Seite 25 beschriebene virtuelle Adapterverbindung ist die Ursache für den serialisierten Ablauf. Sobald ein sendender Adapter A sich reserviert hat, und er erfolgreich den empfangenden Adapter B belegen konnte, bleibt die Verbindung bestehen, bis die Sequenz beendet ist. Dies bedeutet zwar, daß kein dritter Adapter die laufende Übertragung stören kann, es führt jedoch auch dazu, daß nun ausschließlich A mit B kommunizieren kann. Nicht möglich wäre nun parallel eine Übertragung von A nach C, da der Adapter A sich exklusiv für den Adapter B reserviert hat und alle von dem Adapter C kommenden Nachrichten ignorieren würde. Bei großen Dialogsystemen mit

mehreren Benutzern ist es üblich, daß verschiedene Anwender gleichzeitig mit verschiedenen an das Netz angeschlossenen Rechnern und Servern arbeiten möchten. Solche Kommunikationswünsche würden von der HYPERchannel/50 Software (Netex) nacheinander ausgeführt werden, was beispielsweise bei einem File-Transfer zwischen zwei Rechnern mit niedriger Kanalgeschwindigkeit zu spürbaren Wartezeiten für die anderen Übertragungswünsche führen kann. Diese Gefahr wird durch die Netex-Software etwas gelindert, da eine virtuelle Verbindung nur für maximal 32 Kbyte bestehen bleibt. Falls die zu übertragende Datei größer als 32 Kbyte ist, wird sie von Netex aufgespalten und die Teildateien werden nacheinander als unabhängige Message-With-Data-Sequenzen gesendet.

Festzuhalten bleibt, daß kein Zeit-Multiplexing von Übertragungswünschen von der selben Quelle zu unterschiedlichen Zielen stattfinden kann.

7.5 OVERHEAD BEI DER ÜBERTAGUNG MEHRERER DATEIEN

Im Experiment 5 wurde das Verhalten des HYPERchannel/50, der Adapter, der Software und des Hosts untersucht, wenn eine Folge von mehreren Dateien von A nach B übertragen werden soll. Insbesondere sollte beobachtet werden ob ein nahtloser Übergang zwischen zwei Dateiübertragungen möglich ist, oder ob es an diesen kritischen Stellen zu Verzögerungen kommen würde. Mit Sicherheit konnte man schon vorher feststellen, daß die gleiche Datenmenge in einer Datei schneller übertragen werden kann, als dies mit zwei Dateien möglich ist. Als Grund für den Overhead bei zwei Dateien findet man zwei Ursachen:

- die aufwendige Phase der Präperierung und der Reservierung des Partners muß ein zweites Mal durchlaufen werden.
- in dem Zeitraum zwischen dem Ende der 1. Datei und der Übertragung des ersten Blockes der 2. Datei kann keine parallele Aktivität von Kanal und HYPERchannel/50 stattfinden.

Das 5. Experiment wurde in drei Fälle untergliedert:

1. Bei zwei gleich schnellen Kanälen zeigt sich, daß außer der oben genannten Verzögerung keine weitere Verzögerung in Kauf genommen werden muß. Solange der empfangende Adapter noch mit der Abarbeitung des letzten Blockes über seinen angeschlossenen Kanal beschäftigt ist, erhält der sendende Adapter von seinen Host das erste neue Paket. Nachdem nun dieses Paket vollständig im Puffer des Adapters 1 eingetroffen ist, versucht der Adapter 1 den Adapter 2 zu reservieren. Der Adapter 2 ist 32 Mikrosekunden (plus Propagation-Delay) früher freigegeben worden und kann sofort wieder vom Adapter 1 reserviert werden. Der Übergang von einer Datei zu der nächsten, kann aufgrund der Verzahnung von Kanalübertragung und Trunkübertragung mit kaum bemerkbarer Verzögerung stattfinden.
2. Unter der Voraussetzung, daß Host 2 über einen schnellen Kanal verfügt und an Host 3 überträgt, der einen langsameren Kanal besitzt, zeigt sich, daß der Übergang von einer Datei auf die nächste darauffolgende nicht unkritisch ist (Abbildung 33 auf Seite 71). Während der sendende Adapter 2 schon von seinem Host 2 freigegeben wurde, nachdem das letzte Paket übertragen und vom Empfänger bestätigt wurde, bleibt der Adapter 3 noch so lange belegt bis seine beiden Puffer über den Kanal des Hosts 3 geleert werden konnten. Der Adapter 2, der nun eine neue Verbindung mit 3 aufbauen möchte, hat schon von seinem Host über den schnellen Kanal ein neues Paket geschickt bekommen, während Adapter 3 noch mit der Abarbeitung seines alten Paketes über den langsameren Kanal beschäftigt ist, und findet somit den Adapter 3 noch reserviert vor. Erst nachdem der Host 3 den letzten Block der alten Datei quittiert hat, wird der Adapter 3 freigegeben und kann für eine neue Dateiübertragung vom Adapter 2 belegt werden. Es zeigt sich, daß zwischen zwei Dateienübertragungen eine gewisse Zeit verstreicht, in der der Adapter 2 vergeblich versucht den Adapter 3 zu reservieren. Insgesamt läuft die Übertragung dieser beiden Dateien nicht mit maximaler Geschwindigkeit ab, da während der Übermittlung des ersten Pakets der neuen Datei der Adapter 3 kein Paket im Puffer zur Verfügung hat, welches er gleichzeitig über den Kanal an seinen Host 3 schicken könnte.

Event-data= From: 2 To: 3 Host-File 4 Time: 100																		
Event-data= From: 2 To: 3 Host-File 6 Time: 2100																		
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				1.3				1.3
Delay	:	1.0		12		1.0		12		ms/Block				1.5		12		1.5

0	:
500	:	.		..		1 S3		c.	
1000	:	.		..		1 S3		c.	
1500	:	.		..		0 S3		tc		B		t.		0 R2		..		.
2000	:	.		..		0 S3		.c		.		c.		0 R2		..		.
2500	:	.		..		0 A3		.t		B		ct		0 R2		..		.
3000	:	.		..		2 S3		c.		.		cw		0 R2		..		.
3500	:	.		..		2 S3		c.		.		.c		1 R2		..		.
4000	:	.		..		2 S3		t.		B		.c		1 R2		..		.
4500	:	.		..		2 S3		t.		B		.c		1 R2		..		.
5000	:	.		..		2 S3		t.		B	
5500	:	.		..		1 S3		tc		B		t.		0 R2		..		.
6000	:	.		..		1 S3		.c		.		c.		0 R2		..		.
6500	:	.		..		0 S3		ct		B		ct		0 R2		..		.
7000	:	.		..		0 S3		c.		.		cw		0 R2		..		.
7500	:	.		..		0 A3		t.		B		tc		1 R2		..		.
8000	:		wc		1 R2		..		.
8500	:		wc		1 R2		..		.
9000	:		c.		2 R2		..		.
9500	:		c.		2 R2		..		.
10000	:		c.		2 R2		..		.
10500	:

Abbildung 33. Übertragung mehrerer Dateien: Idle

- Falls der Empfänger einen schnelleren Kanal als der Sender besitzt, ergibt sich keine weitere Verzögerung, da der Empfänger-Adapter schon wieder freigegeben wurde, während der Sender-Adapter noch über den Kanal des Hosts 3 gefüllt wird. Sobald der erste Puffer des Adapters 3

mit einem Paket der neuen Sequenz gefüllt ist, probiert der Adapter 3 den Adapter 2 zu reservieren und ist sofort erfolgreich (Abbildung 34 auf Seite 72).

Event-data= From: 3 To: 2 Host-File 4 Time: 100									
Event-data= From: 3 To: 2 Host-File 6 Time: 1100									
TIME :	H1	: A1 :	H2	: A2 :	HC	: A3 :	H3	: A4 :	H4

Speed :	2.0		2.0		Mbyte/s		1.3		1.3
Delay :	1.0	12	1.0	12	ms/Block		1.5	12	1.5

0 :
500 :	c.	1 S2	..	.
1000 :	c.	1 S2	..	.
1500 :	c.	1 S2	..	.
2000 :	.	..	0 R3	t.	B	tc	0 S2	..	.
2500 :	.	..	0 R3	c.	.	.c	0 S2	..	.
3000 :	.	..	0 R3	c.	.	.c	0 S2	..	.
3500 :	.	..	1 R3	.t	B	.t	0 A2	..	.
4000 :	.	..	1 R3	.c	.	c.	2 S2	..	.
4500 :	.	..	1 R3	.c	.	c.	2 S2	..	.
5000 :	c.	2 S2	..	.
5500 :	.	..	0 R3	t.	B	tc	1 S2	..	.
6000 :	.	..	0 R3	c.	.	.c	1 S2	..	.
6500 :	.	..	0 R3	c.	.	.c	1 S2	..	.
7000 :	.	..	1 R3	.t	B	ct	0 S2	..	.
7500 :	.	..	1 R3	.c	.	c.	0 S2	..	.
8000 :	.	..	1 R3	.c	.	c.	0 S2	..	.
8500 :	.	..	2 R3	t.	B	t.	0 A2	..	.
9000 :	.	..	2 R3	c.
9500 :	.	..	2 R3	c.
10000 :

Abbildung 34. Übertragung mehrerer Dateien: Retry

7.6 DEADLOCK AUFLÖSUNG

Das Experiment 6 beschäftigt sich mit der Problematik des Reservierungsverfahrens (siehe Kapitel "Virtuelle Adapter-Verbindungen" auf Seite 25). Es läßt sich zwischen zwei Fällen differenzieren:

1. Falls es sich um einen 'Two-Adapter-Loop' handelt, d.h. falls der Adapter A den Adapter B belegen möchte und quasi gleichzeitig Adapter B den Adapter A reservieren möchte, kann dies vom Protokoll festgestellt werden und die beiden Übertragungen werden serialisiert. Dies ist möglich, da mit jedem Reservierungsversuch die Adresse des Senders mitgeschickt wird. Wenn nun die Adresse des Senders mit der Adresse des zu reservierenden Adapters übereinstimmt, handelt es sich um ein gegenseitiges Reservierungsbemühen. Dieser Fall kann also erkannt und gelöst werden. In Abbildung 35 auf Seite 74 zeigt sich, daß zunächst der Adapter 1 freigegeben wird (Time=1500: 'S'; Time=2000: 'R') und eine Übertragung vom Host 2 an den Host 1 stattfinden kann. Nach einer gewissen Verzögerungsphase wird der Übertragungswunsch des Hosts 1 erneut in die Ereigniswarteschlange eingereiht und kann nun ohne Schwierigkeiten ausgeführt werden (Time=7000: 'S').

Event-data= From: 1 To: 2 Host-File 4 Time: 100																		
Event-data= From: 2 To: 1 Host-File 4 Time: 600																		
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				1.3				1.3
Delay	:	1.0		12		1.0		12		ms/Block				1.5		12		1.5

0	:
500	:	1 S2		c.	
1000	:	1 S2		c.		1 S1		c.	
1500	:	1 S2		t.		1 S1		c.		B	
2000	:	0 R2		t.		0 S1		tc		B	
2500	:	0 R2		c.		0 S1		.c	
3000	:	0 R2		ct		0 A1		.t		B	
3500	:	1 R2		.c	
4000	:	1 R2		.c	
4500	:
5000	:
5500	:
6000	:
6500	:
7000	:	1 S2		c.	
7500	:	1 S2		c.	
8000	:	0 S2		tc		0 R1		t.		B	
8500	:	0 S2		.c		0 R1		c.	
9000	:	0 A2		.t		0 R1		ct		B	
9500	:	.		..		1 R1		.c	
10000	:	.		..		1 R1		.c	
10500	:

Abbildung 35. Two-Adapter-Reservation-Request-Loop

2. Wenn es sich jedoch um einen Reservierungsversuch handelt, an dem 3 oder mehr Adapter beteiligt sind, kommt das Retry-Verfahren zum Zug (Abbildung 36 auf Seite 76). In diesem Beispiel versucht der Adapter 1 den Adapter 2 zu belegen. Adapter 2 möchte Adapter 3 reservieren,

welcher wiederum Adapter 1 als Empfänger belegen möchte. Diese Schleife löst sich erst auf, wenn bei einem Adapter der Retry-Counter abgelaufen ist und er sich selbst freigibt. Nun kann eine Übertragung stattfinden (siehe nochmals "Virtuelle Adapter-Verbindungen" auf Seite 25). In der Tabelle kann man erkennen, daß sich der Adapter 1 freigegeben hat, und, daß daraufhin eine virtuelle Verbindung zwischen Host 3 und Host 1 aufgebaut wird. Der Adapter 1 hat vorher seinen erfolglosen Versuch den gewünschten Adapter zu reservieren an eine höhere Protokollebene (Layer 6) gemeldet, welche diesen Übertragungswunsch nach einer gewissen Wartezeit erneut in die Warteschlange einreißt. In dem Simulationsprogramm wird, wie in dem Kapitel "Modell" auf Seite 35 angedeutet, keine Meldung an eine höhere Protokollebene gegeben (da diese nicht mehr vorhanden ist), sondern der zwangsweise abgebrochene Reservierungsversuch wird nach einer wählbaren Zeitschranke wieder zur Ausführung gebracht.

		Event-data=		From: 1	To: 2	Host-File	6	Time:	100									
		Event-data=		From: 2	To: 3	Host-File	6	Time:	600									
		Event-data=		From: 3	To: 1	Host-File	6	Time:	900									
TIME	:	H1	:	A1	:	H2	:	A2	:	HC	:	A3	:	H3	:	A4	:	H4

Speed	:	2.0				2.0				Mbyte/s				1.3				1.3
Delay	:	1.0		12		1.0		12		ms/Block				1.5		12		1.5

0	:
500	:	2 S2		c.	
1000	:	2 S2		c.		2 S3		c.		.		c.		1 S1		..		.
1500	:	2 S2		t.		2 S3		c.		B		c.		1 S1		..		.
2000	:	2 S2		t.		2 S3		t.		B		c.		1 S1		..		.
2500	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
3000	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
3500	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
4000	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
4500	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
5000	:	2 S2		t.		2 S3		t.		B		t.		1 S1		..		.
5500	:	.		..		2 S3		t.		B		t.		1 S1		..		.
6000	:	.		..		2 S3		t.		B		t.		1 S1		..		.
6500	:	0 R3		t.		2 S3		t.		B		tc		0 S1		..		.
7000	:	0 R3		c.		.		..		B		.c		0 S1		..		.
7500	:	0 R3		c.		.		..		B		.c		0 S1		..		.
8000	:	1 R3		.t		.		..		B		.t		0 A1		..		.
8500	:	1 R3		.c	
9000	:	1 R3		.c	
9500	:

Abbildung 36. Three-Adapter-Reservation-Request-Loop

An dieser Stelle wird deutlich, daß es sinnvoll ist, den Wert des Retry-Counters von der Priorität der Adapters abhängig zu machen. Einerseits würde bei gleichzeitigem Abbruch der Reservierungsbemühungen durch alle Adapter der Konflikt nicht gelöst werden können (siehe Kapitel "Virtuelle Adapter-Verbindungen" auf Seite 25). An-

dererseits könnte dadurch erreicht werden, daß der Adapter mit der kleinsten Priorität zuerst aufgibt und auf diese Weise das Prioritätenverfahren unterstützt wird.

7.7 NUTZEN DES TRUNKS BEI ZWEI PARALLELEN ÜBERTRAGUNGEN

In diesem Versuch sollte der parallele Ablauf zweier unabhängiger Dateien auf dem HYPERchannel/50 untersucht werden (Abbildung 37 auf Seite 78). Während Adapter 1 mit Adapter 2 kommuniziert, besteht zusätzlich eine Verbindung zwischen Adapter 3 und Adapter 4. Die Übertragungsrate des HYPERchannel/50 ist gegenüber den Kanalgeschwindigkeiten der angeschlossenen Rechner so hoch, daß eine quasi parallele Abarbeitung der Sequenzen zweier disjunkter Adapterpaare möglich ist. Dabei kommt es zwar zu Verzögerungen in der Trunk-Ebene (Adapter 1: 't' bei Time=2000,2500), jedoch werden diese innerhalb der gesamten Antwortzeit der Datei durch die Wartezeit auf den Kanal geschluckt.

Diese zeitliche Verzahnung ist sowohl innerhalb einer Message-With-Data Sequenz, als auch zwischen diesen Sequenzen bei einem File-Transfer erlaubt und wird auch durch das Protokoll unterstützt. Diese Fähigkeit des HYPERchannel/50 stellt einen wesentlichen Vorzug als Verbindungsnetzwerk für Großrechner dar.

```

      Event-data= From: 3 To: 4 Host-File 10 Time: 100
      Event-data= From: 1 To: 2 Host-File 12 Time: 600
TIME   : H1   : A1   : H2   : A2   : HC   : A3   : H3   : A4   : H4
-----
Speed  : 2.0           2.0           Mbyte/s           1.3           1.3
Delay  : 1.0   12     1.0   12 ms/Block           1.5   12     1.5
-----
  0 :      .      ..      .      ..      .      ..      .      ..      .
500 :      .      ..      .      ..      .      c.      4 S4      ..      .
1000 : 5 S2      c.      .      ..      .      c.      4 S4      ..      .
1500 : 5 S2      c.      .      ..      .      c.      4 S4      ..      .
2000 : 4 S2      tc      0 R1      t.      B      tc      3 S4      t.      0 R3
2500 : 4 S2      tc      0 R1      t.      B      .c      3 S4      c.      0 R3
3000 : 3 S2      ct      0 R1      ct      B      .c      3 S4      c.      0 R3
3500 : 3 S2      c.      0 R1      cw      B      ct      2 S4      ct      0 R3
4000 : 2 S2      tc      1 R1      tc      B      c.      2 S4      .c      1 R3
4500 : 2 S2      .c      1 R1      wc      .      c.      2 S4      .c      1 R3
5000 : 1 S2      ct      2 R1      ct      B      tc      1 S4      tc      1 R3
5500 : 1 S2      ct      2 R1      ct      B      .c      1 S4      c.      2 R3
6000 : 0 S2      tc      3 R1      tc      B      .c      1 S4      c.      2 R3
6500 : 0 S2      .c      3 R1      wc      B      ct      0 S4      ct      2 R3
7000 : 0 A2      .t      4 R1      ct      B      c.      0 S4      .c      3 R3
7500 :      .      ..      4 R1      cw      .      c.      0 S4      .c      3 R3
8000 :      .      ..      5 R1      .c      B      t.      0 A4      tc      3 R3
8500 :      .      ..      5 R1      .c      .      ..      .      c.      4 R3
9000 :      .      ..      .      ..      .      ..      .      c.      4 R3
9500 :      .      ..      .      ..      .      ..      .      c.      4 R3
10000 :      .      ..      .      ..      .      ..      .      ..      .

```

Abbildung 37. Zwei parallele Übertragungen

In diesem Kapitel werden die während der Untersuchung des HYPERchannel/50 gewonnenen Ergebnisse vorgestellt.

8.1 AUSWIRKUNG UNTERSCHIEDLICHER KANALGESCHWINDIGKEIT

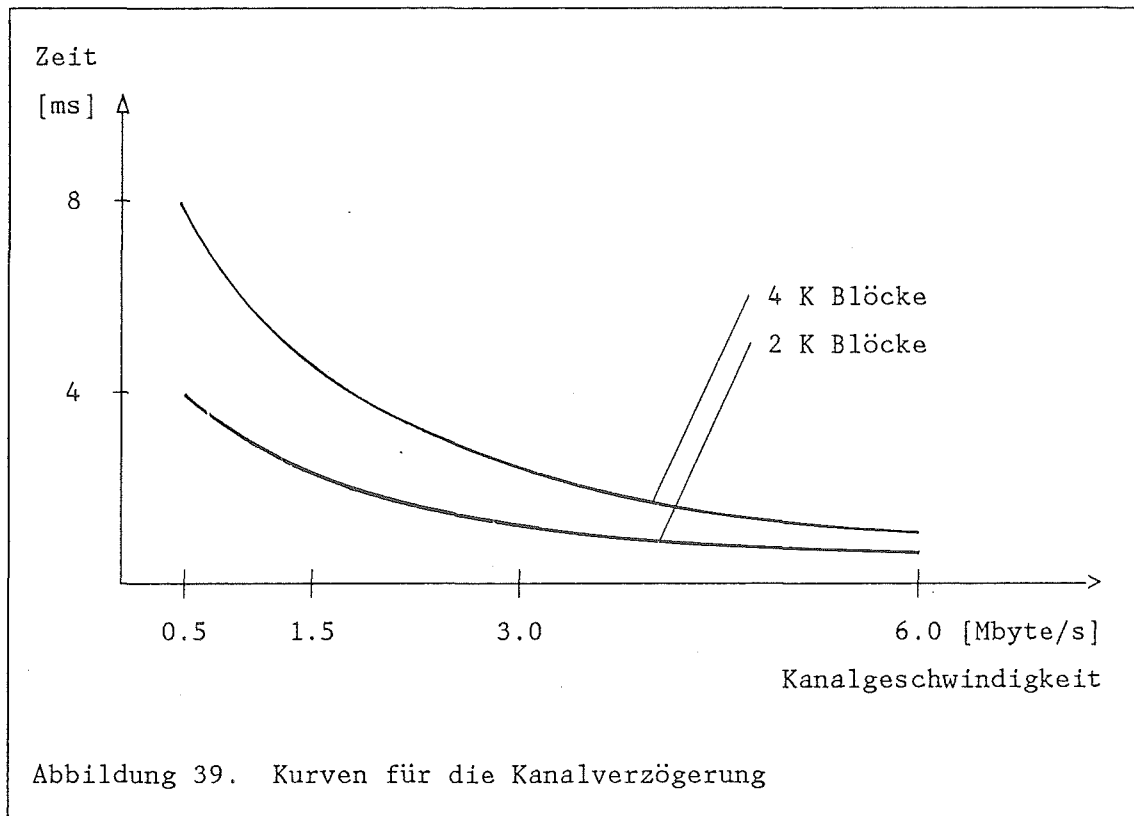
Die in allen Simulationsläufen verwendeten Rechnertypen sind in Abbildung 38 aufgeführt und werden in den Experimenten durch ihre Übertragungsgeschwindigkeit dargestellt. Während es sich bei den IBM Maschinen und der CRAY um die Kanalgeschwindigkeit handelt, wird bei der PDP der HYPERchannel/50-Adapter über einen sogenannten 'Direct-Memory-Access'-Anschluß (DMA) mit dem Rechner verbunden. Für die weiteren Untersuchungen ergeben sich dadurch keine Unterschiede, und es wird hier allgemein von Kanälen gesprochen.

Channel-Speed		Channel-Delay [ms]	
Typ	[Mbyte/s]	2 KB/Block	4 KB/Block
PDP 11/84	0.5	4	8
	1.0	2	4
IBM-3033	1.5	1.33	2.66
	2.0	1	2
IBM-3081	3.0	0.66	1.33
	4.0	0.5	1
CRAY X-MP	6.0	0.33	0.66

Abbildung 38. Kanalgeschwindigkeit <-> Kanalzeit

In Abbildung 38 ist für die einzelnen Rechner neben den Anschlußgeschwindigkeiten die Zeit aufgetragen, welche die Kanäle (DMA) benötigen, um

einen 2 bzw. 4 Kbyte Block zu übertragen. Man kann erkennen, daß die Kanalzeiten (natürlich) mit wachsender Kanalgeschwindigkeit abnehmen (Abbildung 39 auf Seite 80).



Auf Grund der hohen Übertragungsrate des HYPERchannel/50 spielen die Kanalzeiten jedoch eine wichtige Rolle, da sie einen wesentlichen Anteil an der Antwortzeit für eine Datei haben (siehe auch "Gleiche Kanalgeschwindigkeit" auf Seite 61).

8.2 EFFEKTIVE DATENRATE

Dieser Abschnitt beschäftigt sich mit der maximalen effektiven Datenrate, welche die übertragenen Daten pro Zeiteinheit in Megabit pro Sekunde (Mbit/s) angibt, und als ein Kriterium für die Leistungsfähigkeit des HYPERchannel/50 angesehen werden kann. Auf die effektive Datenrate hat die Dateigröße, die Puffergröße der Adapter und die Übertragungsrate des Ka-

nals, an dem der Adapter angeschlossen ist, Einfluß. Es wurden zwei Experimente durchgeführt, wobei in dem ersten Versuch die Kanalgeschwindigkeit unberücksichtigt blieb und in dem zweiten die Kanalgeschwindigkeit durch die Werte existierender Rechner laut Abbildung 38 auf Seite 79 ersetzt wurde. Die Übertragung findet zwischen zwei gleichwertigen Hosts statt, die durch keinen weiteren Dateitransfer gestört werden. Die Prioritäten innerhalb der Hosts haben keine Auswirkung auf die Datenrate, da nur zwei Adapter an dem Trunk angeschlossen sind. Das WFF ist während der Simulationsläufe ausgeschaltet und es wird ein Lauf mit 2 Kbyte Puffern, sowie ein Lauf mit 4 KByte Puffern durchgeführt.

Simulationsprogramm			Franta und Heath			
-----			-----			
			errechnet		simuliert	
Filesize	2 KB	4 KB	2 KB	4 KB	2 KB	4 KB
[KByte]	Blocks	Blocks	Blocks	Blocks	Blocks	Blocks
2	20.5	----	21.37	-----	20.02	-----
4	25.2	29.8	26.05	29.76	26.50	-----
6	27.3	----	28.15	-----	-----	-----
8	28.5	34.2	29.33	34.17	-----	-----
16	29.8	36.9	31.33	36.93	-----	-----
32	30.5	38.4	32.44	38.51	-----	-----
100	32.0	39.5	33.25	39.66	-----	-----
10000	33.3	40.0	33.64	40.22	34.70	41.31

Abbildung 40. Maximale Effektive Datenrate [Mbit/s]

In dem ersten Experiment wurde ein unendlich schneller Kanal angenommen, um die ermittelten Werte mit den Zahlen von Franta und Heath /Fran 84/ vergleichen zu können. Betrachtet wird ein Sender, der Dateien von unterschiedlicher Länge an einen Empfänger schickt. In /Fran 84/ wurde sowohl ein Simulationsmodell, als auch ein analytisches Modell für den

HYPERchannel/50 entwickelt, so daß auf verschiedene Weise erhaltene Werte zum Vergleich zur Verfügung stehen (Abbildung 40).

Wie erwartet steigt mit der Filegröße auch die Datenrate an. Der Grund für diesen Anstieg liegt in der, relativ zu der Anzahl der übertragenen Pakete, geringeren Auswirkung der am Anfang der Message-With-Data-Sequenz notwendigen Reservierungsphase. Je Paket erhält man somit einen geringeren Verwaltungs-Overhead.

Die simulierten Werte stimmen mit den von Franta und Heath angegebenen Zahlen recht gut überein, wobei die für 2 Kbyte Puffer ermittelten Werte etwas unter den in /Fran 84/ angegebenen Zahlen liegen.

Außerdem zeigte sich, daß bei Verwendung von 4 Kbyte Puffern eine höhere Datenrate erzielt wird, als wenn 2 Kbyte Puffer benutzt worden wären. Dieses Verhalten wird verständlich, wenn man den Aufbau eines 4 Kbyte Blocks gegenüber einem 2 Kbyte Block betrachtet. Ein 4 Kbyte Paket enthält zu 92 % Daten, während ein 2 Kbyte Paket 87 % Daten beinhaltet.

a) 2 Kbyte pro Block:

Filesize [KByte]	Kanalgeschwindigkeit [Mbyte/s]				
	PDP	3033	3081	CRAY	
	0.5	1.5	3.0	6.0	∞
2	1.9	4.8	7.8	11.5	20.5
4	2.6	6.9	11.9	17.1	25.2
6	2.9	8.1	14.3	20.4	27.3
8	3.2	8.8	16.0	22.6	28.5
12	3.4	9.7	18.1	25.3	29.2
16	3.6	10.3	19.4	26.9	29.8
32	3.8	11.2	21.7	29.8	30.5
100	3.9	11.7	23.6	32.1	32.0
10000	4.1	12.2	24.5	33.2	33.3

b) 4 Kbyte pro Block:

Filesize [KByte]	Kanalgeschwindigkeit [Mbyte/s]				
	PDP	3033	3081	CRAY	
	0.5	1.5	3.0	6.0	∞
4	1.9	5.1	8.7	13.5	29.8
8	2.6	7.2	12.9	20.2	34.2
12	3.0	8.4	15.3	24.2	35.9
16	3.2	9.1	16.9	26.9	36.9
32	3.6	10.4	20.0	32.2	38.4
100	3.9	11.6	22.9	37.1	39.5
10000	4.1	12.2	24.4	39.7	40.0

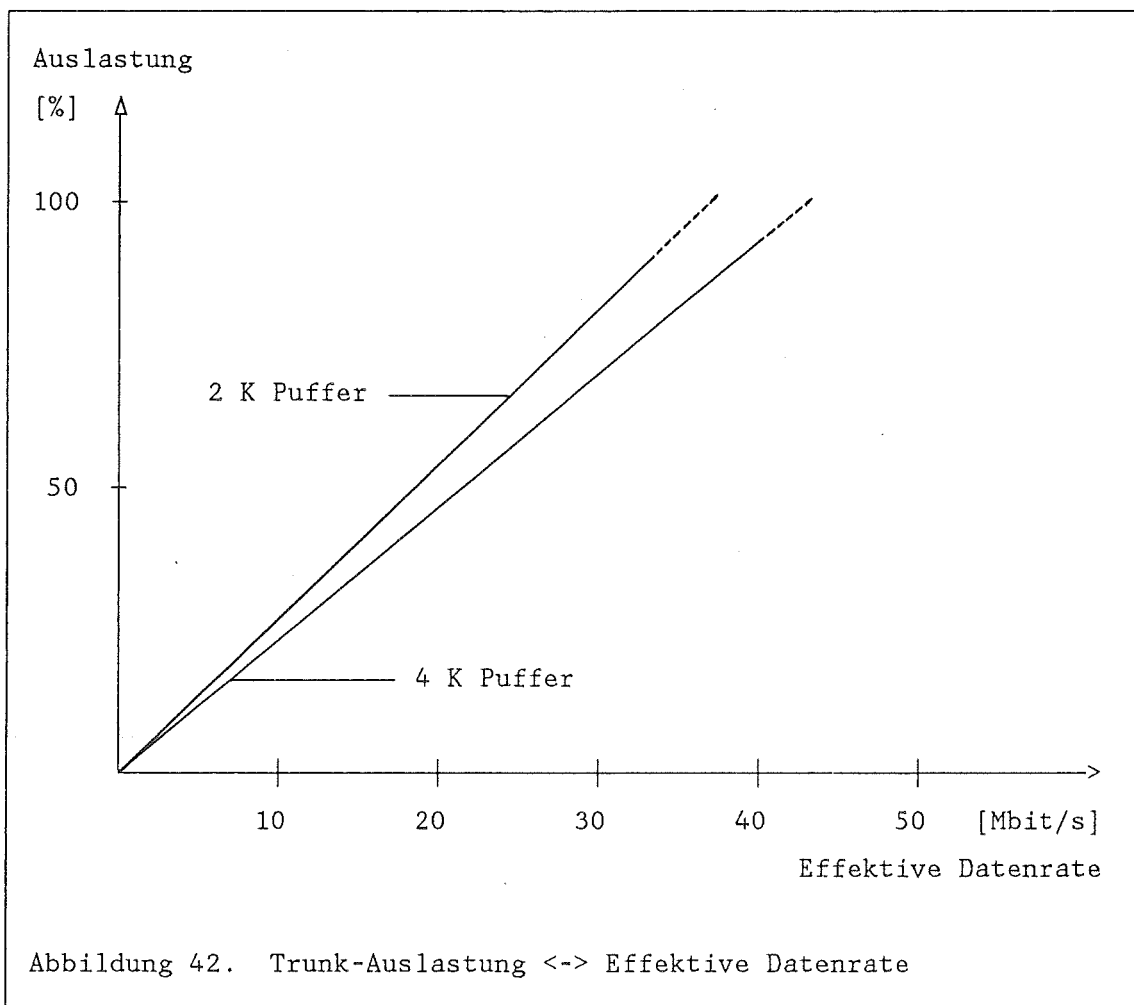
Abbildung 41. Effektive Datenrate [Mbit/s] <-> Kanalgeschwindigkeit

Als Erweiterung des 1. Experiments wird in der 2. Untersuchung der zusätzliche Einfluß unterschiedlicher Anschlußgeschwindigkeiten betrachtet. Es zeigt sich, daß sowohl mit der Zunahme der Kanalgeschwindigkeit, als auch mit dem Anstieg der Dateigröße die Datenrate anwächst. Unverkennbar ist

auch, daß die Kanalgeschwindigkeit des Host einen sehr starken Einfluß auf die Datenrate hat, wobei die von der CRAY erzielten Werte für große Dateien fast die obere Grenze eines imaginären Rechners mit unendlichen Kanalgeschwindigkeit erreicht (Dateigröße 10 Mbyte, 2 Kbyte Puffer, CRAY: 33.2 Mbit/s; 4 Kbyte Puffer, CRAY: 39.7 Mbit/s).

8.3 TRUNK-AUSLASTUNG

Die Abbildung 42 zeigt die Beziehung von Trunk-Auslastung (%) zu effektiver Datenrate (Mbit/s), wobei zwischen 2 Kbyte Puffern und 4 Kbyte Puffern unterschieden wurde.



In diesem Experiment ist das WFF ausgeschaltet und es werden unendlich schnelle Kanäle unterstellt. An den Trunk angeschlossen werden zunächst zwei Adapter, die mit ihren Hosts als ein Paar einen Dateitransfer durchführen, dann findet aber auch eine gleichzeitige Übertragung von 4 bzw. 6 Adaptern statt. Die durchgezogene Linie entspricht den Werten, die durch ein Adapterpaar erzielt werden, während die Werte der gestrichelten Linie nur durch zwei oder mehr Adapterpaare erreicht werden können.

Man kann auch hier erkennen, daß eine höhere Datenrate erzielt wird, wenn 4 Kbyte Puffer verwendet werden. Deutlich wird außerdem, daß ein Adapterpaar den Trunk nicht vollständig auslasten kann (2 Kbyte Puffer: 87%; 4 Kbyte Puffer: 92%), und daß die erzielbare Datenrate noch unter dem maximal möglichen Wert liegt (2 Kbyte Puffer: 33.3 Mbit/s; 4 Kbyte Puffer: 40.0 Mbit/s). Erst wenn mehrere Übertragungen gleichzeitig stattfinden, werden die Maximalwerte erreicht (2 Kbyte Puffer: 38.1 Mbit/s; 4 Kbyte Puffer: 43.2 Mbit/s).

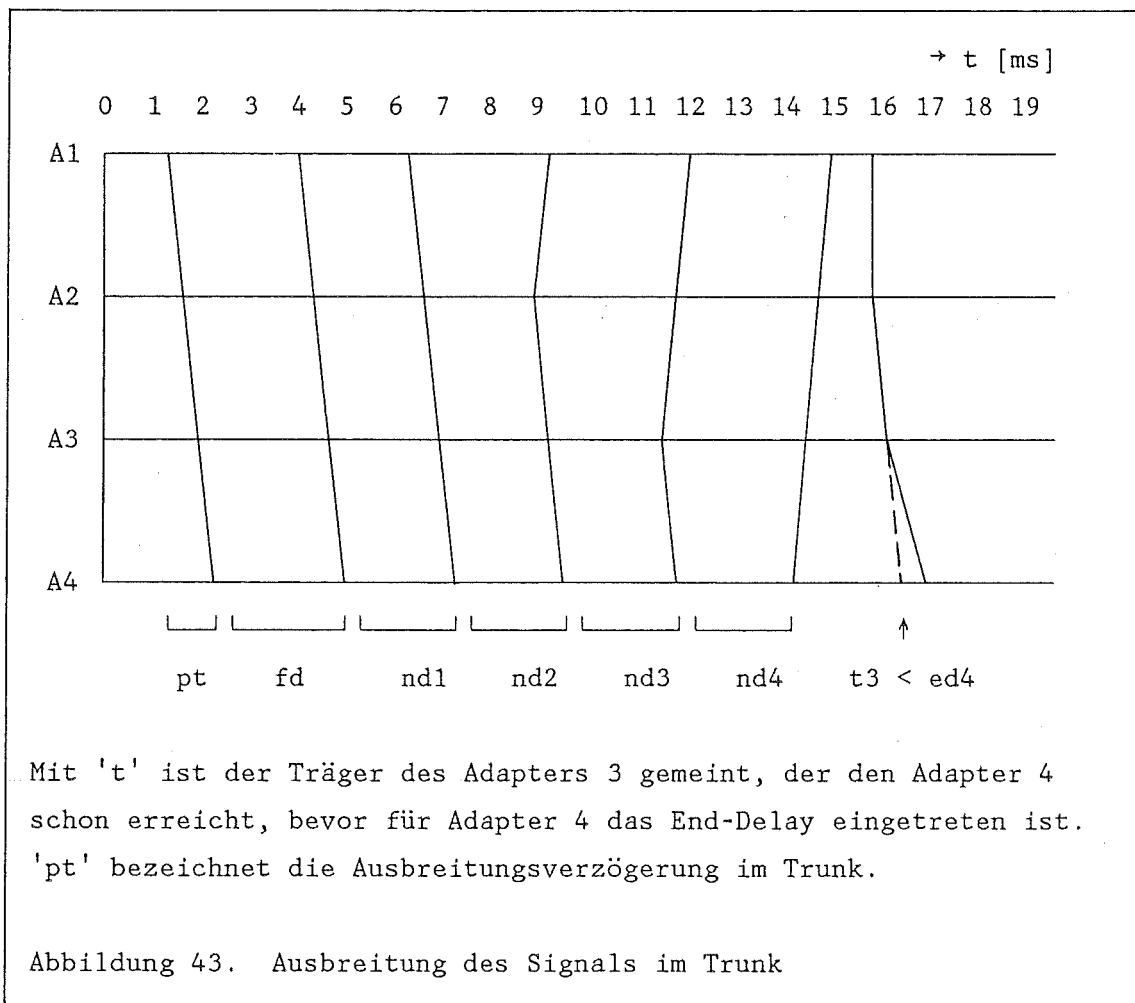
Bei dieser Abbildung läßt sich auch recht deutlich der Verwaltungs-Overhead des Protokolls, sowie die Belastbarkeit des Trunks abschätzen. Ein Adapterpaar mit 2 Kbyte Puffern und unendlicher Kanalgeschwindigkeit kann nur 66.6% der verfügbaren Kapazität nutzen, während bei Verwendung von 4 Kbyte Puffern dieser Wert auf 76.2% steigt, erreicht er bei mehreren Paaren und 4 Kbyte Puffern einen Höchstwert von 86.4%.

8.4 END-DELAY

Das im HYPERchannel/50-Protokoll beschriebene Verfahren zur Berechnung des End-Delays führt zu einem unterschiedlichen Eintritt der Adapter in die Contention-Phase (siehe "Protokoll" auf Seite 13). Das End-Delay eines Adapters wird durch den Abstand zu dem von ihm aus am weitesten entfernten Adapter bestimmt; je größer dieser Abstand ist, desto größer ist das End-Delay des Adapters (Abbildung 43 auf Seite 86). Daraus folgt, daß ohne Berücksichtigung der 'Propagation Time' die mehr in der Mitte des Trunks angeschlossenen Adapter kleinere End-Delays besitzen (Adapter 2 und 3:

15µs), als die an den beiden Enden positionierten Adapter (1 und 4: 16µs). Die mittleren Adapter können ihr kleineres End-Delay bei einem neuen Übertragungswunsch nutzen und die Übertragung beginnen, während die äußeren Adapter sich noch in der N-Delay Phase befinden. Diese ungleiche Errechnung des End-Delay führt dazu, daß die mittleren Adapter bevorzugt werden, da:

1. sie zuerst die Contention-Phase erreichen,
2. das WFF der äußeren Adapter nicht zurückgesetzt wird (dies geschieht nur durch Eintritt des End-Delays, siehe "Wait-Flip-Flop" auf Seite 25). Während der nächsten N-Delay Phase dürfen diese Adapter in ihren Windows, aufgrund ihres immer noch gesetzten WFF, keine Übertragung starten.



Es zeigt sich somit, daß bei hoher Last und aktiviertem WFF die mittleren Adapter durch das verwendete Verfahren zur Errechnung der End-Delays bevorzugt werden. Der Trunk-Zugriff wird fairer, wenn das End-Delay statt:

$$ed_k = nd_n + 2 * pt * d_{k,f(k)} + 1600 \text{ ns}$$

mit:

$$ed_k = nd_n + 2 * pt * d_{1,n} + 1600 \text{ ns}$$

berechnet wird. Dabei gibt $d_{1,n}$ die Gesamtlänge des Trunks an und ist unabhängig von k (d.h. für alle Adapter gleich).

Die folgende Abbildung 44 auf Seite 88 zeigt eine Konfiguration mit sechs Hosts und ihren Adaptern, die alle gleichzeitig eine Datei von 100 Kbyte übertragen möchten. Es wurden zwei Simulationsläufe durchgeführt, wobei zwischen originalem und modifiziertem End-Delay unterschieden wurde. Die Last wurde erhöht, indem die Kanalgeschwindigkeiten der angeschlossenen Rechner vergrößert wurden. Aus diesem Grund ergibt sich bei hoher Last (Fall a) auch eine geringere Antwortzeit, als bei mittlerer Last (b). In dem Fall a) konnte für ausgeschaltetes WFF keine repräsentativen Werte gemessen werden, da das dritte Paar keinen Zugang zum Trunk erhielt.

a) Hohe Last (Wait-Flip-Flop ON):

SENDER - Statistik	Datei-Antwortzeit [ms]		
	Host 1	Host 2	Host 3
Original End-Delay	65.6	65.2	66.0
Geändertes End-Delay	65.3	65.7	66.1

b) Mittlere Last:

SENDER - Statistik	Datei-Antwortzeit [ms]		
	Host 1	Host 2	Host 3
Wait-Flip-Flop OFF	67.2	67.6	68.1
Original End-Delay (Wait-Flip-Flop ON)	68.8	68.3	69.2
Geändertes End-Delay (Wait-Flip-Flop ON)	67.2	67.7	68.2

Abbildung 44. Modifiziertes End-Delay

Experimente mit dem modifizierten End-Delay ergaben, daß die Wartezeit für die Adapter nun in der Reihenfolge ihrer Priorität geordnet sind (Abbildung 44, hohe Last: 65.3 → 65.7 → 66.1). Die in /Fran 82/ beobachtete Zunahme der Wartezeit, welche mit der Zunahme der Kollisionen bei modifiziertem End-Delay aufgrund der garantierten Zurücksetzung des WFF und damit der erhöhten Anzahl von Kollisionen begründet wurde, konnte in den durchgeführten Versuchen nur in geringem Maße bei sehr hoher Last festgestellt werden.

Entscheidend ist jedoch, daß das geänderte End-Delay über den gesamten Lastbereich in der Lage ist, eine Ordnung bzgl. der Prioritäten der Adapter herzustellen. Zusätzlich zeigte sich, daß die Wartezeiten der

Adapter bei modifiziertem End-Delay und geringer bis mittlerer Last kleiner ausfallen, als bei standardmäßigen End-Delay und sie fast die Zeiten der gleichen Konfiguration bei ausgeschalteten WFF erreichen (siehe Abbildung 44 auf Seite 88 ; die Dateiantwortzeit ändert sich für die Hosts der Reihe nach von 'WFF off' zu 'WFF on' (aber modifiziert) wie folgt: 67.2 → 67.2, 67.6 → 67.6, 68.1 → 68.2).

8.5 ANTWORTZEIT

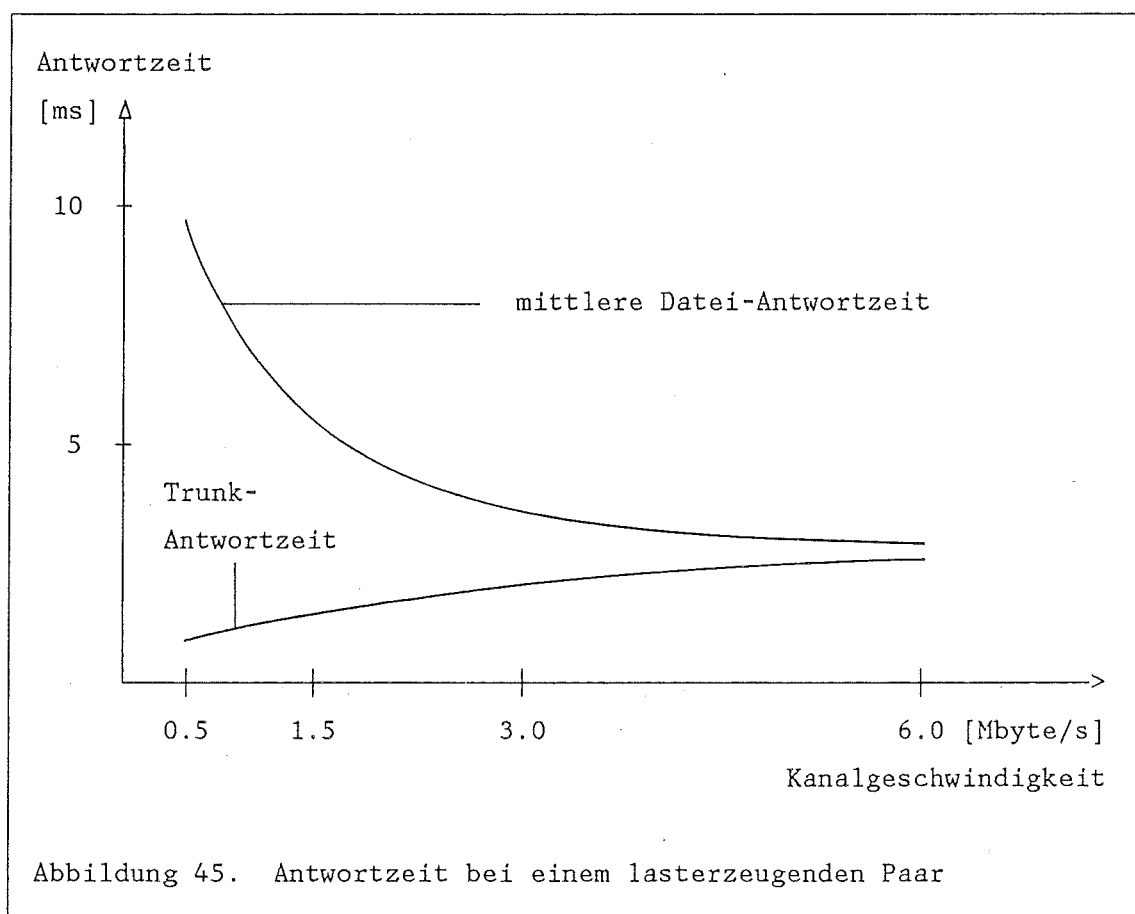
In den beiden folgenden Experimenten wird die Antwortzeit für eine Datei bei unterschiedlicher Belastung des HYPERchannel/50 untersucht. Die Versuche sind so aufgebaut, daß die Hosts mit den höheren Prioritäten durch Übertragung einer großen Datei eine Grundlast schaffen, während die mittlere Antwortzeit mehrerer kleiner Dateien eines Hosts mit niedriger Priorität betrachtet wird. Als gemessene Host werden nacheinander die Rechner PDP 11/84, IBM 3033, IBM 3081 und CRAY X-MP betrachtet, die stellvertretend für die Anschlußgeschwindigkeit gemäß der Abbildung 38 auf Seite 79 stehen.

Die beiden Experimente unterscheiden sich durch die Anzahl der lasterzeugenden Hosts.

a) Belastung erzeugt von einem Adapterpaar

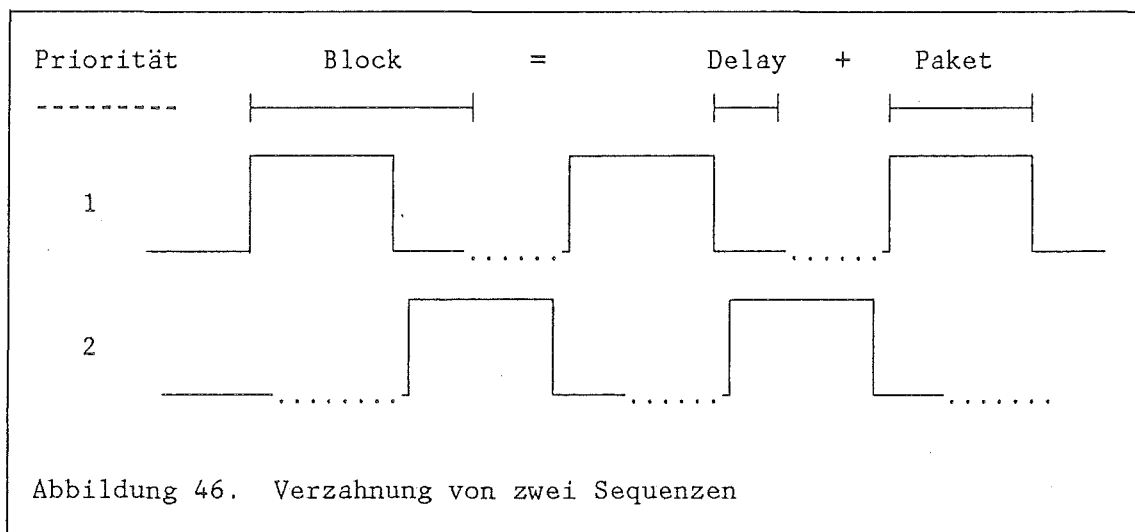
In dem ersten Versuch wurden vier an den HYPERchannel/50 angeschlossene Rechner untersucht, wobei der Host 1 an den Host 2 ein File von 100 Kbyte schickt, während Host 3 an Host 4 vier Files mit je 4 Kbyte überträgt. Für das Experiment ist das Wait-Flip-Flop ausgeschaltet und es werden 2 Kbyte Puffer verwendet. Insgesamt wurden 16 Simulationsläufe durchgeführt, da für jeden der vier gemessenen Rechnertypen nacheinander auch die vier Rechner als lasterzeugende Hosts eingesetzt werden. Betrachtet wird sowohl die insgesamt notwendige Zeit für einen Dateitransfer, als auch die reine Trunkzeit für die Datei. Die 32 durch die Simulation ermittelten

Werte wurden in Abbildung 45 auf Seite 90 zu Kurven approximiert. Dabei ist an der Y-Achse die Wartezeit in Millisekunden aufgetragen und an der X-Achse die Kanalgeschwindigkeiten der gemessenen Hosts. Die Kurven zeigen die 'File-Response-Time' bzw. 'Trunk-Response-Time', gemäß der Definition in Abbildung 23 auf Seite 53, des gemessenen Adapterpaares mit der niedrigeren Priorität. Die an der X-Achse aufgetragenen Kanalgeschwindigkeiten entsprechen der Rate des lasterzeugenden Paares mit der hohen Priorität.



Die obere Kurve stellt die mittlere Datei-Antwortzeit, und die untere Kurve die Trunk-Antwortzeit des gemessenen Paares dar. Es zeigt sich, daß bei nur zwei angeschlossenen Adapterpaaren keine wesentliche Wartezeit für das Paar mit der niedrigeren Priorität entsteht. Dies gilt sogar dann, wenn das erste Paar mit einer sehr hohen oder sogar unendlichen Kanalgeschwindigkeit die Daten an den Trunk schickt. Der Grund für dieses Verhalten liegt in dem Aufbau eines Blocks, da jeder Block aus dem eigentlichen Paket und einem Delay besteht (Abbildung 46 auf Seite 91). Während des

Delays gibt der Adapter den Trunk frei und verrichtet eigene Arbeiten. Der Sender mit der niedrigeren Priorität kann diese Phase nutzen und sein eigenes Paket abschicken. Das hat zur Folge, daß der Adapter mit der höchsten Priorität und der Adapter mit der zweithöchsten gleich oft den Trunk erhalten. Durch den verzahnten Ablauf der beiden Sequenzen werden beide Files gleich stark verzögert und das Wait-Flip-Flop hat keinen Einfluß auf die Prioritäten, da eine Übertragung abwechselnd stattfindet und das WFF jedesmal zurückgesetzt wird, nachdem beide einmal dran waren.



Aus einer solchen Konstellation folgt, daß die Antwortzeit eines Files im wesentlichen durch die Kanalgeschwindigkeit des angeschlossenen Hosts bestimmt wird. Zwar steigt die benötigte Trunkzeit bei wachsender Belastung des HYPERchannel/50 durch die höhere Kanalgeschwindigkeit, jedoch wird dieser Zuwachs durch die geringere Verzögerung der Kanalübertragung kompensiert, sodaß für Kanalgeschwindigkeiten über 3 Mbyte/s die File-Response-Time nahezu konstant bleibt.

b) Belastung erzeugt von zwei Adapterpaaren

In dem zweiten Versuch sollte durch Hinzufügen eines 2. lasterzeugenden Adapterpaares die Voraussetzung für die Untersuchung folgender Eigenschaften des HYPERchannel/50-Protokolls geschaffen werden:

1. Wirkungsweise des WFF;

2. Ablauf des Prioritätenverfahrens;
3. "Belastbarkeit" des HYPERchannel/50;

Bei dieser Konfiguration können gleichzeitig drei Übertragungen stattfinden, wobei:

- Adapter 1 an Adapter 2 ein File mit 160 Kbyte überträgt;
- Adapter 3 an Adapter 4 ein File mit 160 Kbyte überträgt;
- Adapter 5 an Adapter 6 vier Files mit je 4 Kbyte sendet;

Die Prioritäten entsprechen den Nummern der Adapter und als Puffergröße wurden 2 Kbyte gewählt. Während des Experiments wird die mittlere Antwortzeit der vier kleinen Dateien betrachtet, die in Abhängigkeit der unterschiedlichen lasterzeugenden Rechner bestimmt wird. Es wurden sechs Versuche durchgeführt, wobei schrittweise die Rechnerkonfiguration geändert wird (Abbildung 47).

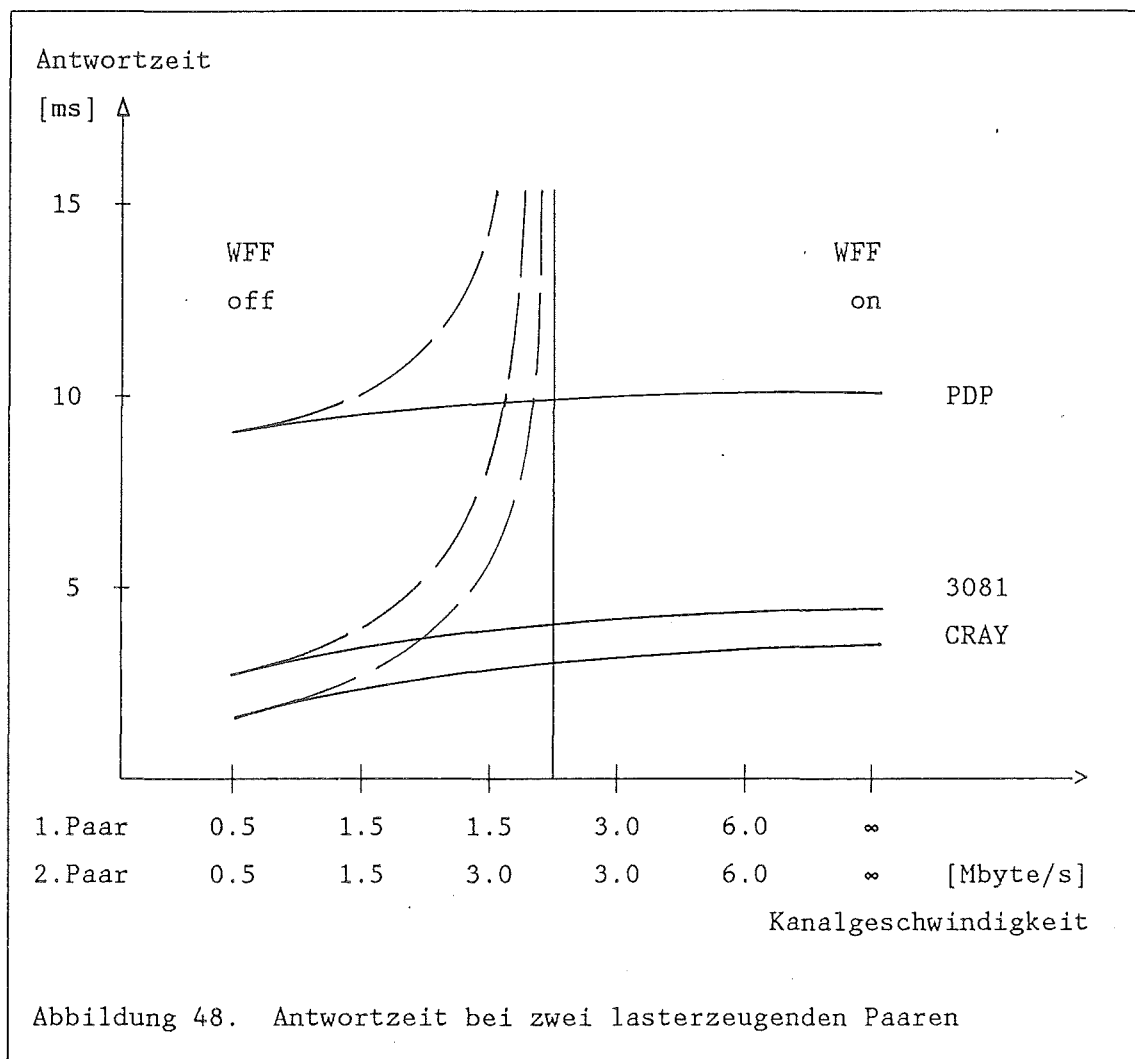
Rechner Type	Kanal [Mbyte/s]	Nummer des Versuchs					
		1.	2.	3.	4.	5.	6.
PDP 11/84	0.5	2 Paare					
IBM-3033	1.5	2 Paare 1 Paar					
IBM-3081	3.0	1 Paar 2 Paare					
CRAY X-MP	6.0	2 Paare					
	∞	2 Paare					

Abbildung 47. Aufbau der Versuche zur Messung der Antwortzeit

Zwei unterschiedliche Konfigurationen werden betrachtet:

1. WFF ON
2. WFF OFF

In Abbildung 48 auf Seite 93 sind die in der Simulation gewonnenen Werte durch Kurven approximiert worden.

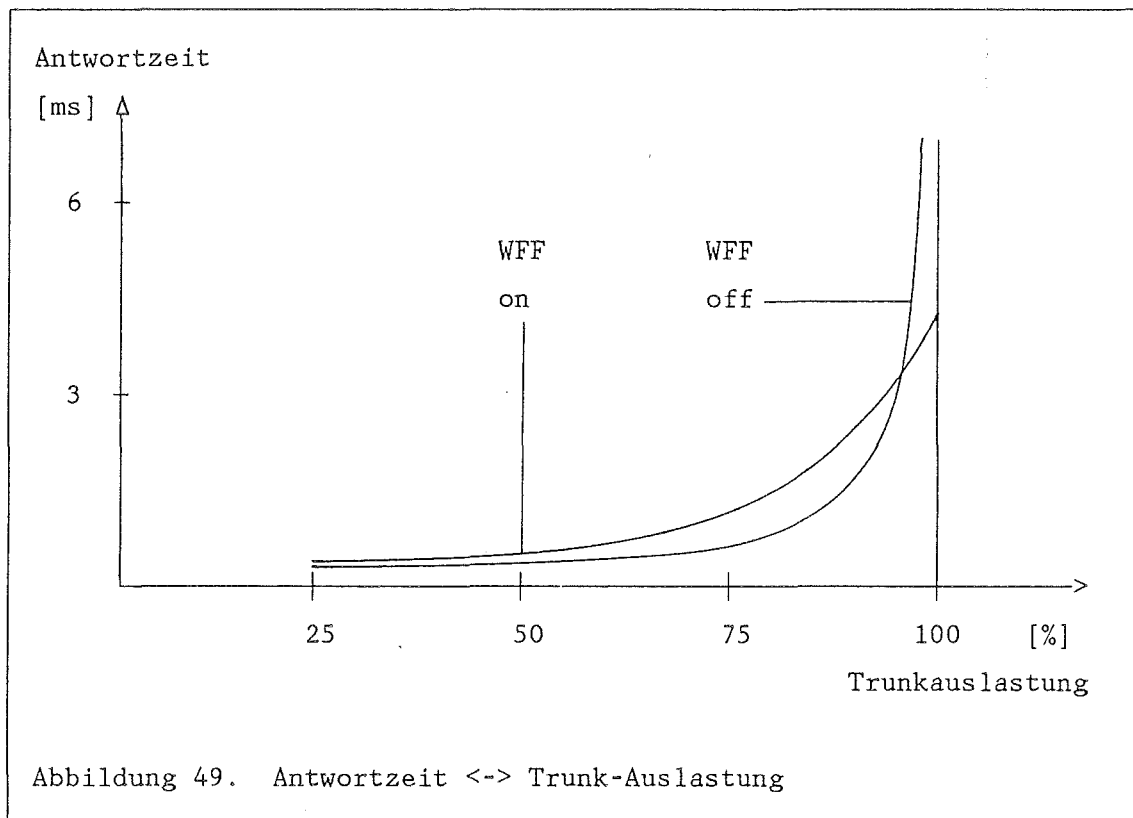


Die gestrichelt aufgetragenen Kurven geben die mittlere Antwortzeit für eine Datei ('File-Response-Time', siehe Abbildung 23 auf Seite 53) bei ausgeschalteten WFF's wieder. Man kann deutlich erkennen, daß die Antwortzeit bei zunehmender Trunk-Belastung stark ansteigt, und ab einer bestimmten Rechnerkonfiguration gegen unendlich wächst. Diese Grenze liegt bei zwei lasterzeugenden Rechnerpaaren bei einer Kanalgeschwindigkeit von 3.0 Mbyte/s und 2.2 Mbyte/s. Sobald die beiden Paare mit den höchsten Prioritäten diese Werte überschreiten, erhält bei ausgeschaltetem WFF kein weiterer Adapter mehr Zugang zum Trunk. Der Grund für dieses Verhalten liegt in der Verzahnung der beiden höchstwertigen Sequenzen, die zusammen, wegen ihren hohen Kanalgeschwindigkeiten, den HYPERchannel/50 ganz auslasten und den niederwertigen Adaptern keine Möglichkeit geben ihr File zu übertragen (Abbildung 46 auf Seite 91).

Bei aktiviertem WFF zeigt sich, daß die Kurven für die mittlere Datei-Antwortzeit zunächst parallel zu den bei deaktiviertem WFF erhaltenen Graphen verlaufen, um dann aber nur ganz leicht anzusteigen. Das Verhalten bedeutet, daß selbst bei unendlicher Kanalgeschwindigkeit der lasterzeugenden Hosts die Antwortzeit bei eingeschaltetem WFF für ein File nicht beliebig wächst, sondern nur ganz leicht zunimmt und endlich ist.

An dieser Stelle erkennt man, daß das CSMA/CA-Verfahren deutlich von dem Ethernet-Protokoll abweicht, da zwar in der Regel die mittlere Last bei Ethernet relativ gering ist /Scho 80/, es jedoch bei hoher Last nicht möglich ist eine obere Schranke für die Wartezeit anzugeben.

Die Wartezeit bei dem HYPERchannel/50 wird bei eingeschaltetem WFF im wesentlichen durch die Anzahl der Adapter bestimmt, die in diesem Moment eine Übertragung durchführen möchten. Das WFF wird nach jeder Sendung gesetzt und bewirkt, daß jeder Adapter reihum einmal die Chance hat sein Paket zu senden, bevor es bei allen Adaptern wieder zurückgesetzt wird (siehe Kapitel "Wait-Flip-Flop" auf Seite 25). Bei diesem Round-Robin Verfahren wird somit die Wartezeit eines Adapters durch die Anzahl der Übertragungswünsche der anderen Adapter bestimmt, da sie in ihren Windows ihr Übertragungsrecht wahrnehmen können oder nicht. Falls sonst kein Adapter senden möchte, verstreicht das N-Dealy und nach dem End-Dealy wird das WFF zurückgesetzt, so daß das nächste Paket geschickt werden kann. Falls jeder Adapter ein Paket zu senden hat, wird jeder sein Fenster während des N-Delays ausnutzen und erst nachdem alle einmal gesendet haben, darf der Adapter mit der höchsten Priorität sein nächstes Paket übertragen.



Um die Auswirkungen des WFF auf die Wartezeiten noch genauer analysieren zu können, ist in Abbildung 49 die Antwortzeit des HYPERchannel/50 ('Trunk-Response-Time') aufgetragen. Diese Zeit ist für ein Adapterpaar mit niedriger Priorität bei wachsender Trunk-Belastung durch zwei Adapterpaare mit hoher Priorität gemessen worden.

Die Puffergröße der Adapter beträgt 2 Kbyte und die Priorität wird nicht geändert. Es wurden zwei Simulationsläufe durchgeführt, wobei zwischen ein- und ausgeschaltetem WFF unterschieden wurde.

Man kann erkennen, daß die Kurve für die Trunk-Antwortzeit bei ausgeschaltetem WFF und anwachsender Trunkauslastung gegen unendlich steigt, und die Ursache der rasch zunehmenden Datei-Antwortzeit (WFF off) in Abbildung 48 auf Seite 93 ist. Die Antwortzeit des Trunks bei ausgeschaltetem WFF steigt bei dieser Konfiguration von sechs Adapter bis zu einem Maximalwert von 3.3 ms. Die beiden Kurven schneiden sich bei ca. 97% Trunkauslastung und bis zu diesem Wert liegt die Antwortzeit des Trunks bei aktiviertem WFF oberhalb der Antwortzeit des Trunks bei deaktiviertem WFF. Die größte Differenz ist allerdings sehr gering und beträgt 0.4 ms.

In diesem Versuch wird deutlich, daß auf das WFF bei hoher Last und bei großer Anzahl der angeschlossenen Adapter nicht verzichtet werden darf.

In dieser Arbeit wurde zunächst der HYPERchannel/50 untersucht und das verwendete Protokoll beschrieben. Danach wurde ein Modell eingeführt, mit dessen Hilfe die OSI-Ebenen 1 bis 5 abgebildet werden können. Als nächstes wurde das Modell in ein Simulationsprogramm umgesetzt und die daraus gewonnenen Erkenntnisse vorgestellt.

Das Simulationsprogramm verfügt über die folgenden Freiheitsgrade:

- Konfiguration:
 - Anzahl der Adapter;
 - Prioritäten der Adapter;
 - Positionierung der Adapter am Trunk;
 - Kanalgeschwindigkeit der Hosts;
- Parameter:
 - Aufbau der Message-With-Data-Sequenz;
 - Puffergröße des Adapters;
 - Schaltung des Wait-Flip-Flops;
 - Größe des Retry-Counters;
 - Wahl des End-Delays;
- Last:
 - Sende-Adapter;
 - Empfangs-Adapter;
 - Startzeit;
 - Dateigröße;

Im Gegensatz zu der von W.R. Frata und J.R. Heath durchgeführten Simulation /Fran 84/, wurde keine Konfiguration mit mehreren physischen Leitungen ('Multi-Trunk-Network') betrachtet, jedoch konnten zusätzlich die folgenden Punkte untersucht werden:

1. Auswirkung der Host-Adpterverbindung (Kanal);
2. Verhalten bei Kollisionen ('Reserve-Loops');

3. Effekte der virtuellen Adapterverbindung ('Half-Duplex', kein Zeit-Multiplexing);

Die durchgeführten Versuche lassen sich zu folgenden Ergebnissen zusammenfassen:

- Die Berechnung des End-Delays sollte so modifiziert werden, daß sie unabhängig von der Position der Adapter am Trunk ist, damit die mittleren Adapter nicht bevorzugt werden und eine Ordnung bzgl. der Prioritäten hergestellt wird.
- Das WFF wirkt sich positiv erst bei sehr hoher Last oder bei einer großen Anzahl der angeschlossenen Adaptern aus, jedoch ist es dann unverzichtbar.
- Bei aktiviertem WFF erhält man bei einem Datei-Transfer je Block (2 Kbyte) Antwortzeiten die im Bereich von 10 ms liegen und sehr zufriedenstellend sind (Das Positionieren des Schreibkopfes bei einer Magnetplatte beträgt etwa 16-20 ms).
- Die Adapterpuffergröße sollte 4 Kbyte betragen, falls der überwiegende Anteil der Trunk-Belastung durch File-Transfer hervorgerufen wird.
- Das Limit für den Retry-Counter sollte nicht zu hoch gewählt werden (ca. 50-100), damit Reserve-Loops schneller aufgelöst werden können.
- Die Hosts mit der größten Kanalgeschwindigkeit sollten die höchsten Prioritäten bekommen. (Die Rechner mit der geringeren Kanalgeschwindigkeit können eine Trunk-Verzögerung mit weniger Einbußen überstehen, da sie auf den langsamen Kanal warten.)
- Die Adapter sollten in der Reihenfolge ihrer Priorität an den Trunk angeschlossen werden.

Die Untersuchungen des HYPERchannel/50 mit Hilfe eines Simulationsprogramms führten zu dem Ergebnis, daß das verwendete Protokoll, die gewählte Übertragungsgeschwindigkeit sowie die zugrunde lie-

gende Hardware prinzipiell in der Lage ist, die Maschinen eines großen Rechenzentrums, unter der Randbedingung zufriedenstellender Antwortzeiten und hoher Durchsatzleistungen, zu verbinden.

10.0 LITERATUR

- /Burr 83/ Burr W.E.
*An Overview of the Proposed American National Standard for
Local Distributed Data Interfaces*
National Bureau of Standards, Washington, D.C.
Communications of the ACM, August 1983, Vo.26, No.8
- /Carp 84/ Carpenter R.
A comparison of two 'guaranteed' local network access methods
National Bureau of Standards, Washington, D.C.
Data Communications, February 1984, Page 143-152
- /Chan 81/ Chandy K.M., Sauer C.H.
Computer Systems Performance Modeling
Prentice Hall International Inc. ISBN 0-13-1651-75-7
- /Chri 78/ Christensen G.S., Franta W.R.
*Design and Analysis of the Access Protocol for Hyperchannel
Networks*
3rd USA-Japan Computer Conference, 1978, Page 86-93
- /Ethe 80/ Digital Equipment, Intel, Xerox Corporations
*The Ethernet: A Local Area Network, Data Link Layer and Physical
Layer Specifications.*
Version 1.0, 30 September 1980
- /Donn 79/ Donnelley J.E., Yeh J.W.
*Interaction between Protocol Levels in a Prioritized CSMA
Broadcast Network*
Journal of Computer Networks, Vol.3, No.1, März 79
- /Fran 80/ Franta W.R., Bilodeau M.B.
Analysis of a Prioritized CSMA Protocol Based on Staggered Delays
Acta Informatica, Vol.13, No.4, June 1980, Page 299-324

- /Fran 82/ Franta W.R., Heath J.R.
Measurement and Analysis of Hyperchannel Networks
 University Minnesota, Institut of Technologie, Minneapolis, 1982
- /Fran 84/ Franta W.R., Heath J.R.
Performance of Hyperchannel Networks: Parameters, Measurements, Models and Analysis
 IEEE Transactions of Computers, C-33, No.3, March 1984, 249-260
- /Fran 85/ Franta W.R., Heath J.R.
A Simple Model of the Hyperchannel Local Network Channel Selection Procedure in Multi-channel Networks
 University of Southern Maine, Departement of Computer Science
 Portland, 1985
- /Hugh 84/ Hughes J.
Continuous Protokoll for Long Distance Communication Links
 Network Systems Corporation, Minnesota
 Proceedings of Cray User Group Oct 1984, San Francisco
- /Jans 83/ Janson P., Svobodova L., Maehle E.
Filing And Printing Service On A Local Area Network
 IBM Zürich Research Laboratory, Rüschlikon, Switzerland
 ACM, 1983
- /KFA 84/ *Statusbericht 1984*
 Zentralinstitut für angewandte Mathematik der Kernforschungs-
 anlage in Jülich, 1984
- /Katk 81/ Katkin, Raphael, Sprung J.
Simulating a Cable Bus Network in a Multi-computer and Large-scale Application Enviroment
 Proceedings Sixth Conference on Local Networks, 1981
- /Kauf 84/ Kauffels F.J.
Lokale Netze
 DV-Praxis, Köln, 1984

- /Mc1a 84/ McLarty T., Collins W., Devaney M.
A Functinal View of the Los Alamos Central File System
 Sixth IEEE Symposium on Mass Storage Systems
 June 4-7, 1984, Vail, Colorado
- /Mard 84/ Mardsen B.W.
A Standard Pascal Event Simulation Package
 Department of Computer Science, Manchester University, UK
 Software - Practice and Experience, Vol.14(7), July 1984
- /Meis 85/ Meister B.W., Janson P.A., Svobodova L.
*Connection-Oriented versus Connectionless Protocols:
 A Performance Study*
 IEEE Transaction on Computers, Vol.C-34, No.12, Dez 1985
- /Ness 81/ Nessel D.M.
*Hyperchannel Archtekture: A Case Study of some Inadequacies
 in the ISO-OSI Reference Model*
 Lawrence Livermore Laboratory, April 1981
- /Nsc 85/ Network System Corporation
Hyperchannel, System Description Manual
 Publication Number NS 306, 1985
- /Nsc 81/ Network System Corporation
Nucleus Adapter, Description and Theory of Operation
 Reference Manual No. 42990003, 1981
- /Shoc 80/ Shoch J.F., Hupp J.P.
Measured Performance of an Ethernet Local Network
 Communications of the ACM, Vol.23, No.12, 1980
- /Sere 85/ Seret D.
*Sujet de la these: Evalution de performance de protocoles de
 communication - application aux reseaux locaux et aux trans-
 missions pas sattelite*
 These de doctorat d'etat, Universite Pierre et Marie Curie
 Paris

- /Span 81/ Spaniol O.
*Analysis and Performance Evaluation of HYPERchannel Access
Protocols*
Performance Evaluation 1 (1981), 170-179
- /Wats 79/ Watson W.B.
*Simulation of Traffic Dependent Performance of a Prioritized
CSMA Broadcast Network*
Proceedings Fourth Conference on Local Networks, 1979

